

Kepstrální analýza a lftrování lidské řeči

Cepstral analysis and liftering of human speech

Zadání bakalářské práce

Student:

Adam Vůjtek

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

1103R031 Výpočetní matematika

Téma:

Kepstrální analýza a lifierování lidské řeči
Cepstral analysis and liftering of human speech

Zásady pro vypracování:

Kepstrum je anagramem slova spektrum, liftr pak slova filtr. Komplexní kepstrum je definováno jako Fourierova transformace komplexního logaritmu amplitudového spektra signálu, často se též nazývá spektrum spektra. Tzv. power kepstrum je pak definováno jako kvadrát amplitudového spektra Fourierovy transformace logaritmu kvadrátu amplitudového spektra signálu. Kromě analýzy seismických signálů se využívá hlavně k určování fundamentálních frekvencí lidské řeči. Vhodnost toto přístupu spočívá ve snadné separabilitě hlasových efektů plynoucí z aditivity logaritmů jejich power kepstra tzv. quefrency domény (tyto efekty se chovají konvolučně v časové a multiplikativně ve frekvenční oblasti). Často využívanou vlastností je, že konvoluci signálů lze vyjádřit jako součet jejich kepster. Liftr je pak filtr operující na kepstru.

Cílem této práce je se seznámit s problematikou kepstrální analýzy, implementovat potřebné funkce včetně algoritmu FFT a porovnat její použití a efektivitu s klasickou spektrální analýzou lidské řeči.

Cepstrum is an anagram of the word spectrum, lifter then of the word filter. The complex cepstrum is defined as the Fourier transform of complex logarithm of the signal's amplitude spectrum, it is often called the spectrum of the spectrum. The power cepstrum is then defined as the square of the amplitude spectrum of the Fourier transform of the logarithm of square of the signal's amplitude spectrum. Except the seismic signals' analysis it is mostly used to determine the fundamental frequencies of human speech. The suitability of this approach consists in easy separability of the vocal effects resulting from the logarithm's additivity of their power cepstrum in quefrency domain (these effects convolve in the time domain and multiply in frequency domain). The often used feature is the signals' convolution equals to their cepstra sum. The lifter is filter operating on the cepstrum.

The aim of this thesis is the study of the cepstral analysis, the implementation of required functions including the FFT algorithm and the comparison of its use and efficiency with the classical spectral analysis of the human speech.

Seznam doporučené odborné literatury:

- [1] B. P. Bogert, M. J. R. Healy, and J. W. Tukey, The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo Autocovariance, Cross-Cepstrum and Saphe Cracking, Proceedings of the Symposium on Time Series Analysis (M. Rosenblatt, Ed) Chapter 15, 209-243. New York: Wiley, 1963.
- [2] A. V. Oppenheim, Superposition in a class of nonlinear systems (Ph.D. dissertation), Res. Lab. Electronics, Massachusetts Institute of Technology, Cambridge, MA, 1965.
- [3] A. Michael Noll (1964), Short-Time Spectrum and Cepstrum Techniques for Vocal-Pitch Detection," Journal of the Acoustical Society of America, Vol. 36, No. 2, pp. 296-302.
- [4] T.J.Moir and J.F.Barrett, A kepstrum approach to filtering, smoothing and prediction with application to speech enhancement, Proc Royal Society A, Vol.459,2003, pp.2957-2976.
- [5] D. Horák, Diskrétní transformace, <http://mi21.vsb.cz/modul/diskretni-transformace>, 2011.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. David Horák, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013

doc. RNDr. Jiří Bouchala, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

V Ostravě 1. května 2013

.....
Najdu

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 1. května 2013

.....
Najdu

Na tomto místě bych chtěl poděkovat především svému vedoucímu bakalářské práce panu Ing. Davidu Horákovi, Ph.D., který měl se mnou dostatek trpělivosti, aby mi pomohl dovést tuto práci ke zdárnému konci. Také bych rád poděkoval své rodině, především rodičům, kteří to mnohdy se mnou neměli lehké, a přátelům, že mě po celou dobu studia podporovali.

Abstrakt

Cílem této práce bylo seznámit se s problematikou keprální analýzy a liftování řeči. Za tímto účelem jsem implementoval algoritmy a funkce, které jsou potřeba pro výpočet a analýzu spektra a kepra, zejména pak varianty algoritmu FFT, který je základem obou analýz. V této práci porovnávám efektivitu jednotlivých variant a nejlepší z nich pak používám ke zpracování zvukových signálů.

Klíčová slova: liftování lidské řeči, keprální analýza, FFT, DFT, spektrum, filtr, frekvenční analýza

Abstract

The aim of this thesis is the study of cepstral analysis and liftering of human speech. For this purpose I have implemented algorithms and functions that are needed for the calculation and analysis of the spectrum and cepstrum, especially variants of the FFT algorithm, which is the basis for both analysis. In this study, I compare the efficiency of FFT variants and the best of them I use to process audio signals.

Keywords: Liftering of human speech, Cepstrum analysis, FFT, DFT, spectrum, filter, frequency analysis

Seznam použitých zkratk a symbolů

$F(f)$	–	Fourierova transformace
FFT	–	Rychlá Fourierova transformace
DFT	–	Diskrétní Fourierova transformace
FT	–	Fourierova transformace
STFT	–	Oknenní Fourierova transformace
x, v, j, k	–	Proměnné
i	–	Imaginární jednotka
$f(x), D(x), f[x]$	–	Funkce
<i>kurzíva</i>	–	Definice
$d(n)_{j=0}^{\infty}, D(n)_{j=0}^{\infty}$	–	Posloupnosti
\Re	–	Reálný obor hodnot

Obsah

1	Úvod	3
2	Spojité FT	4
2.1	Definice spojité FT	4
2.2	Vlastnosti FT	4
3	Diskrétní FT (DFT)	7
3.1	Definice DFT	7
3.2	Rychlá FT (FFT)	7
3.3	Jak pracuje FFT	9
4	Algoritmy a implementace FT	12
4.1	Algoritmus DFT	12
4.2	Algoritmus dopředné FFT	12
4.3	Algoritmus zpětné FFT	13
4.4	Algoritmus pomocí Daniel-Lanczosovy metody	14
4.5	Paralelní FFT	15
5	Kepstrální analýza	16
5.1	Historie kepstrální analýzy	16
5.2	Aplikace kepstra	17
5.3	Kepstrální pojmy	17
5.4	Liftrvání kepstra	17
5.5	Konvoluce signálů	17
5.6	Charakter řečového signálu	17
5.7	Model vytváření řeči	18
5.8	Reálné kepstrum	19
5.9	Výpočet reálného spektra	19
5.10	Homomorfická filtrace	21
6	Liftering	24
6.1	Low-time liftering	24
6.2	High-time liftering pro odhad intervalu	24
7	Numerické experimenty	27
7.1	Porovnání rychlosti algoritmů FFT	27
7.2	Spektrální analýza české abecedy	28
7.3	Reálná kepstra	34
7.4	Analýza písmene „a“	36
7.5	Analýza slova „abeceda“	39
7.6	Analýza slova „láska“	42
7.7	Aplikace Power kepstra	46

8 Závěr	50
9 Reference	51

1 Úvod

V 21. století lidstvo touží po co nejpohodlnějším ovládní svých mobilních zařízení, tabletů a počítačů. Jednou z možností je ovládní hlasem. Toto je ale velmi problematické, protože každý člověk má jiný tón hlasu, výšku hlasu a také barvu hlasu. Jak najít optimální nástroj pro řešení tohoto problému se pokusím nastínit v této bakalářské práci. V kapitolách 2-4 se budeme zabírat Fourierovou transformací (FT), protože je základem keprální analýzy. Budu se snažit vysvětlit pomocí jakých metod převést hlasový záznam tak, aby byl dobře počítačově čitelný, jaké existují numerické metody výpočtu Fourierovy transformace. V dalších částech se budu věnovat keprální analýze, teorii rozpoznávání lidského hlasu a metodám kterých se využívá v této problematice.

2 Spojitá FT

2.1 Definice spojité FT

FT je vyjádření časově závislého signálu pomocí harmonických signálů, které reprezentují funkce $\sin x$ a $\cos x$. Slouží pro převod z časové oblasti do oblasti frekvenční [1]. Signál může být jak ve spojitém tak i v diskrétním čase. *Definice: FT $S(\omega)$ funkce $s(t)$ je definována integrálním vztahem*

$$S(\omega) = \int_{-\infty}^{\infty} s(t)e^{-i\omega t} dt.$$

Funkci $s(t)$ vypočteme z $S(\omega)$ inverzní FT

$$s(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega)e^{i\omega t} d\omega.$$

Ne vlastní integrály chápeme ve smyslu Cauchyovy hlavní hodnoty tj.

$$\int_{-\infty}^{\infty} [.] d = \lim_{T \rightarrow \infty} \int_{-T}^T [.] d.$$

Dvojice ve FT se nazývají originál (v našem případě $s(t)$) a obraz ($S(\omega)$). Vztah mezi originálem a obrazem má následující tvar [2]

$$S(\omega) = F[s(t)] \text{ a } s(t) = F^{-1}[S(\omega)].$$

V technické oblasti $S(\omega)$ představuje spektrum signálu $s(t)$. Spektrum je komplexní veličina a lze ji vyjádřit ve tvaru

$$S(\omega) = |S(\omega)| e^{i \arg S(\omega)}.$$

Velikost $|S(\omega)|$ nazýváme amplitudové spektrum a úhel $\arg S(\omega)$ fázové spektrum signálu.

2.2 Vlastnosti FT

2.2.1 Věta o linearitě

Věta: Lineární kombinaci signálů odpovídá lineární kombinace jejich spekter [7]

$$F \left[\sum_i c_i s_i(k) \right] = \sum_i c_i F[s_i(k)].$$

2.2.2 Věta o změně měřítka (Podobnost)

Věta: Má-li signál $s(t)$ spektrum $S(\omega)$, má signál $s(at)$, $a \neq 0$ spektrum

$$\frac{1}{|a|} S\left(\frac{\omega}{a}\right).$$

2.2.3 Věta o substituci (Věta o kmitočtovém posunu)

Věta: [7]

$$F\{f(t)e^{i\omega t}\} = F(i\omega - i\alpha)$$

$$\alpha \in \mathbb{R}.$$

2.2.4 Posunutí

Věta:

$$F\{f(t - \tau)\} = F(i\omega e^{-i\omega\tau}),$$

kde $\tau \in \mathbb{R}$ je reálná konstanta. Mění se jen fázové spektrum [7].

2.2.5 Věta o obrazu derivace

Věta:

$$F\{f'(t)\} = i\omega F\{f(t)\} = i\omega F(i\omega),$$

$$F\{f^{(n)}(t)\} = (i\omega)^{(n)} F\{f(t)\} = (i\omega)^{(n)} F(i\omega),$$

kde $\lim_{x \rightarrow \pm\infty} F(i\omega) = 0$ [7].

2.2.6 Derivace obrazu

Věta: [7]

$$F\{-itf(t)\} = \frac{dF(i\omega)}{d\omega}, F\{(-it)^n f(t)\} = \frac{d^n F(i\omega)}{d\omega^n}.$$

2.2.7 Věta o obraze integrálu

Definice: [7]

$$F\left\{\int_{-\infty}^t f(\tau)d\tau\right\} = \frac{1}{i\omega} F(i\omega).$$

2.2.8 Obraz konvoluce dvou funkcí

Věta: Necht' funkce $f, g \in L^2(\mathbb{R})$, pak konvoluce těchto funkcí a její obraz jsou dány následujícími vztahy [7]

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau,$$

$$F \{ (f * g)(t) \} = F(i\omega)G(i\omega),$$

kde platí:

$$F \{ f(t) \} = F(i\omega), F \{ g(t) \} = G(i\omega).$$

2.2.9 Věta o součinu originálů

Věta:

$$F \{ f(t)g(t) \} = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(i\omega_1)G(i\omega - i\omega_1)d\omega_1 = \frac{1}{2\omega} (FG(i\omega)),$$

kde $\omega_1 \in (-\psi, \omega > .$

2.2.10 Parsevalova rovnost (Pro FT)

Věta: [7]

$$\int_{-\infty}^{\infty} |f(t)|^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |F(i\omega)|^2 d\omega = \frac{1}{\pi} \int_0^{\infty} |F(i\omega)|^2 d\omega.$$

2.2.11 Spektrum reálného signálu

Je-li signál reálný, pak pro jeho spektrum platí [1]:

- amplitudové spektrum je sudou funkcí,
- fázové spektrum je lichou funkcí,
- spektrum sudého signálu je sudou reálnou funkcí,
- spektrum lichého signálu je lichou ryze imaginární funkcí.

3 Diskrétní FT (DFT)

3.1 Definice DFT

Jedním nejméně vhodným řešením, jak implementovat na počítači FT je využít klasickou DFT.

Definice: Necht' $f = (f_0, f_1, f_2, \dots, f_{N-1}) \in l^2(N)$ a systém $\phi_n = (\phi_{n,0}, \dots, \phi_{n,N-1})$, kde $\phi_{n,k} = e^{i\frac{2\pi}{N}nk}$ tvoří ortogonální bázi prostoru $l^2(N)$. Přímá DFT vektoru f je definována jako vektor $c = (c_0, \dots, c_{N-1})^T$, jehož n -tá komponenta (koeficient n -té harmonické) je dána vztahem [7]

$$F_n = \langle f, \phi_n \rangle = \sum_{k=0}^{N-1} f_k e^{-i\frac{2\pi}{N}nk}$$

a zpětná DFT vektoru c je definována jako vektor f , jehož n -tá komponenta je dána vztahem

$$f_n = \sum_{k=0}^{N-1} c_k \frac{1}{\sqrt{N}} e^{i\frac{2\pi}{N}nk}$$

Maticový zápis: Označíme $w = e^{-i\frac{2\pi}{N}}$ a $w^{nk} = e^{-i\frac{2\pi}{N}nk}$ pak transformační matice vypadá

$$M_F = \begin{pmatrix} w^0 & w^0 & w^0 & \dots & w^0 \\ w^0 & w^1 & w^1 & \dots & w^{N-1} \\ w^0 & w^2 & w^4 & \dots & w^{2(N-1)} \\ w^0 & w^{N-1} & w^{2(N-1)} & \dots & w^{(N-1)(N-1)} \end{pmatrix}$$

M_F je tvořená ortonormální soustavou funkcí na intervalu $[0, (N-1)\Delta t]$, kde $\Delta t = t_{k+1} - t_k$. Přímá DFT je pak dána vztahem: $F = M_F \cdots f$, a zpětná je dána vztahem $f = \frac{1}{N} M_F^T \cdots F$ [7].

3.2 Rychlá FT (FFT)

FFT patří mezi ty nejefektivnější algoritmy pro výpočet DFT a její inverze. Využívá se v mnoha odvětvích ať už v digitálním zpracování signálu, řešení parciálních diferenciálních rovnic a nebo násobení velkých čísel. FFT má složitost výpočtu pouze $O(N \log N)$ [2]. Jelikož inverzní DFT se pouze liší ve znaménku v exponentu a v koeficientu $1/N$, pak se dá tento algoritmus využít i pro počítání inverzní DFT.

3.2.1 Cooley-Tukey algoritmus

Jednou z možných cest, jak aplikovat FFT je kupříkladu Cooley-Tukey algoritmus. Tento algoritmus využívá metody rozděl a panuj, který rekurzivně zjednodušuje DFT s velikostí složeného čísla do menších DFT o velikostech N_1 a N_2 . V praxi tento algoritmus funguje tak, že transformace se v každém kroku rozdělí na dvě stejně velké transformace velikosti $N/2$. I když původní algoritmus je rekurzivní, ty nejrychlejší algoritmy se této možnosti vyhýbají, protože explicitní rekurze nebývá rychlejší než řešení cyklem (při

velkém počtu opakování). Jelikož Cooley-Tukey algoritmus se dělí do dalších menších DFT, je možné ho zkombinovat s jiným algoritmem [2].

Implementace FT

Vycházíme z DFT dané vztahem

$$F_n = \sum_{k=0}^{N-1} f_k e^{-i \frac{2\pi}{N} nk}, w = w_N = e^{-i \frac{2\pi}{N}}$$

Rozdělením vektoru $f = (f_0, f_1, \dots, f_{N-1})^T$ majícího N složek na 2 vektory o $N/2$ složkách $y = (f_0, f_2, \dots, f_{N-2})^T$ a $z = (f_1, f_3, \dots, f_{N-1})^T$ a to tím způsobem, že vezmeme členy se sudým indexem $y_k = f_{2k}$ a členy s lichým indexem $z_k = f_{2k+1}$, pak lze F_n obrazu $F = (F_0, F_1, \dots, F_{N-1})^T$ rozepsat jako

$$F_n = \sum_{k=0}^{\frac{N}{2}-1} (y_k w^{2nk} + z_k w^{(2k+1)n}), n = 0, 1, \dots, N-1 [7].$$

V důsledku symetrie komplexních čísel můžeme psát:

$$w^{2nk} = e^{-i \frac{2\pi}{N/2} nk}.$$

Obecně platí, že $w_N^{pq} = w_{N/q}^P$, potom můžeme napsat, že

$$F_n = \sum_{k=0}^{N/2-1} y_k w_{N/2}^{nk} + w_N^n \sum_{k=0}^{N/2-1} z_k w_{N/2}^{nk} = Y_n + w_N^n Z_n, n = 0, 1, \frac{N}{2} - 1 [7],$$

kde $Y = DFT(y)$ a $Z = DFT(z)$.

První polovinu koeficientů F_n získáme pomocí DFT vektorů poloviční velikosti. Druhou polovinu získáme jako

$$F_{n+\frac{N}{2}} = Y_n + w_N^{n+N/2} Z_n = Y_n + w_N^n w_N^{N/2} Z_n = Y_n - w_N^n Z_n, n = 0, 1, 2, N/2 - 1$$

protože $w_N^{N/2} = e^{-i \frac{2\pi}{N} \frac{N}{2}} = e^{-i\pi} = -1$. Štěpení opakujeme dokud nedostaneme 2 vektory o jedné složce

$$DFT(y_0) = Y_0 = \sum_{k=0}^0 y_0 w_N^0 = y_0, DFT(z_0) = Z_0 = \sum_{k=0}^0 z_0 w_N^0 = z_0.$$

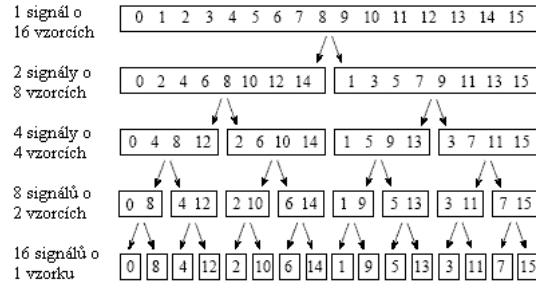
Maticový zápis FFT tedy bude vypadat

$$F = M_{F,n} \cdot f = \begin{pmatrix} I_{\frac{N}{2}} & D_{\frac{N}{2}} \\ I_{\frac{N}{2}} & -D_{\frac{N}{2}} \end{pmatrix},$$

kde $I_{\frac{N}{2}}$ je jednotková matice, diagonální matice $D_{\frac{N}{2}} = \frac{1}{\sqrt{\frac{N}{2}}} \text{diag}(1, w_{2r}^1, w_{2r}^2, \dots, w_{2r}^{N/2-1})$,

další krok bude

Obrázek 1: Ukázka motýlkového algoritmu [3]



$$Y = \begin{pmatrix} I_{\frac{N}{4}} & D_{\frac{N}{4}} \\ I_{\frac{N}{4}} & -D_{\frac{N}{4}} \end{pmatrix} \begin{pmatrix} Y' \\ Z' \end{pmatrix} = B_{\frac{N}{2}} \begin{pmatrix} Y' \\ Z' \end{pmatrix}, Z = \begin{pmatrix} I_{\frac{N}{4}} & D_{\frac{N}{4}} \\ I_{\frac{N}{4}} & -D_{\frac{N}{4}} \end{pmatrix} \begin{pmatrix} Y'' \\ Z'' \end{pmatrix} = B_{\frac{N}{2}} \begin{pmatrix} Y'' \\ Z'' \end{pmatrix},$$

kde $I_{\frac{N}{4}}$ je jednotková matice, diagonální matice $D_{\frac{N}{4}} = \frac{1}{\sqrt{\frac{N}{4}}} \text{diag}(1, w_{2r}^2, w_{2r}^4, w_{2r}^{N/4-2})$ obě řádu $\frac{N}{4} \times \frac{N}{4}$.

V dalším kroku vektory Y', Z', Y'', Z'' znovu rozdělíme na vektory se sudým a lichým indexem, přeindexujeme a několikrát dalším zopakováním dostaneme

$$F = B_N \begin{pmatrix} B_{\frac{N}{2}} & 0 \\ 0 & B_{\frac{N}{2}} \end{pmatrix} \begin{pmatrix} B_{\frac{N}{4}} & 0 & 0 & 0 \\ 0 & B_{\frac{N}{4}} & 0 & 0 \\ 0 & 0 & B_{\frac{N}{4}} & 0 \\ 0 & 0 & 0 & B_{\frac{N}{4}} \end{pmatrix} \dots \begin{pmatrix} B_2 & 0 & 0 & 0 & \dots & 0 \\ 0 & B_2 & 0 & 0 & \dots & 0 \\ 0 & 0 & B_2 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & B_2 \end{pmatrix} P^T f,$$

kde P je permutační matice vytvořená pomocí motýlkového algoritmu (sudo-liché přeskupování sloupců jednotkové matice) [7].

3.3 Jak pracuje FFT

Časová i frekvenční část obsahují jeden signál skládající se z K komplexních čísel. Každé komplexní číslo je tvořeno složkou reálnou a imaginární. Kupříkladu vzorek $X[42]$ se skládá z reálné části $X[42]$ a imaginární části $X[42]$. FFT pracuje na bázi, že signál o N vzorcích v časové sféře se rozloží na N signálů v časové sféře (každý signál je složen z jednoho vzorku). Druhým krokem vypočteme frekvenční spektrum N odpovídajících těmto N signálům. Poté je N spekter sloučeno do jediného frekvenčního spektra. Na obrázku 1 jde vidět jak tento algoritmus pracuje. Signál se rozkládá na K signálů, až dosáhne pouze 1 vzorku. Vzorky v signálu jsou uspořádány takto z jednoho důvodu. FFT převrátí bity všech čísel v binární soustavě, tím pádem dostane úplně novou řadu čísel, než byla původní. Kupříkladu místo dvojky zapsané binárně (0001) po převrácení se stane číslo osm (1000), pětka (0101) se stane desítkou (1010). Na obrázku (obr. 2)

Obrázek 2: Ukázka motýlkového algoritmu na binárních číslech [3]

Vzorky v normálním pořadí			Vzorky po převrácení bitů	
decimálně	binárně		decimálně	binárně
0	0000	⇒	0	0000
1	0001		8	1000
2	0010		4	0100
3	0011		12	1100
4	0100		2	0010
5	0101		10	1010
6	0110		6	0100
7	0111		14	1110
8	1000		1	0001
9	1001		9	1001
10	1010		5	0101
11	1011		13	1101
12	1100		3	0011
13	1101		11	1011
14	1110		7	0111
15	1111		15	1111

můžete vidět princip převrácení bitů. V dalším kroku FFT je třeba naleznout frekvenční spektrum jednoho bodu signálu časového úseku. Frekvenční spektrum je přímo úměrné bodu časového úseku, tzn. v tomto kroku není třeba nic dělat. V tomto případě však platí, že každý signál o jednom vzorku je frekvenčním spektrem a už není signálem časového úseku. Nyní je třeba udělat reverzní operaci, kterou jsme dělali na začátku a to spojit frekvenční spektrum přesně v zaměněném pořadí, než jsme dělali na začátku algoritmu. V jednoduchém popisu to znamená, že ze šestnácti frekvenčních spekter o jednom vzorku vznikne osm frekvenčních spekter o 2 vzorcích, poté se vytvoří čtyři frekvenční spektra o čtyřech vzorcích a v posledním kroku vznikne frekvenční spektrum o šestnácti vzorcích. Názorně je to předvedeno na obr. 3.

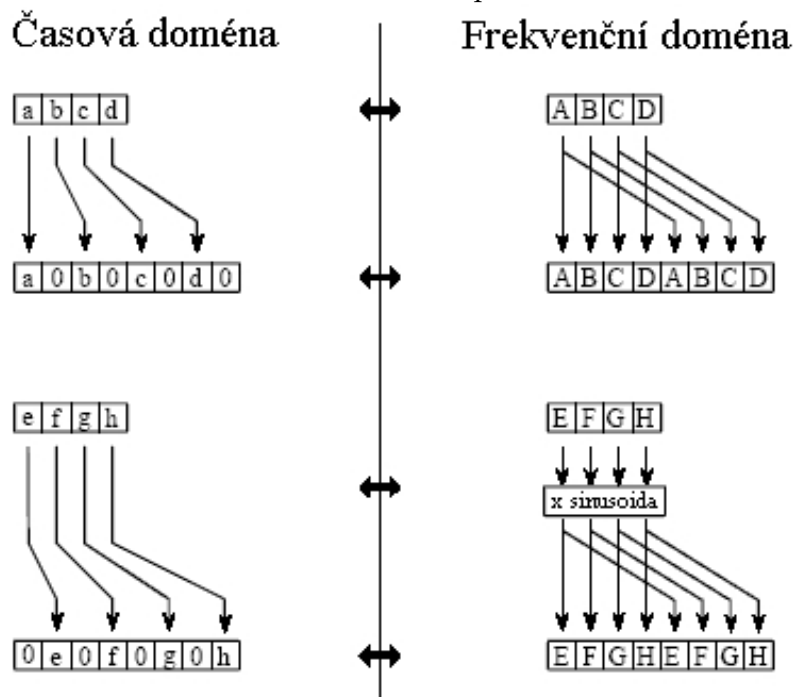
3.3.1 Implementace Daniel-Lancozsova algoritmu

Obecný zápis tohoto algoritmu je dán jako [5]

$$\begin{aligned}
 {}^n F_k &= \sum_{j=0}^{n-1} e^{\frac{-2\pi i k j}{n}} f_j = \\
 &= \sum_{j=0}^{\frac{n}{2}-1} e^{\frac{-2\pi i k (2j)}{n}} f_{2j} + \sum_{j=0}^{\frac{n}{2}-1} e^{\frac{-2\pi i k (2j+1)}{n}} f_{2j+1} = \\
 &= \sum_{j=0}^{\frac{n}{2}-1} e^{\frac{-2\pi i k j}{N/2}} f_{2j} + \omega_n^k \sum_{j=0}^{n/2-1} e^{\frac{-2\pi i k j}{n/2}} f_{2j+1} = {}^{n/2} F_k^e + \omega_n^k \cdot {}^{N/2} F^0 k,
 \end{aligned}$$

kde ${}^{N/2} F_k^e$ je $N/2$ je bodová DFT sudých prvků z originální DFT vyhodnocených s periodou k . Úplně stejně rozdělíme i druhou část pro liché prvky z DFT.

Obrázek 3: Reverzní operace [3]



Obrázek 4: Ukázka algoritmu

$$\begin{aligned}
 {}^8F_5 &= {}^4F_5^e + \omega_8^5 \cdot {}^4F_5^o \\
 &= ({}^2F_5^{ee} + \omega_4^5 \cdot {}^2F_5^{eo}) + \omega_8^5 \cdot ({}^2F_5^{oe} + \omega_4^5 \cdot {}^2F_5^{oo}) \\
 &= (({}^1F_5^{eee} + \omega_2^5 \cdot {}^1F_5^{eeo}) + \omega_4^5 \cdot ({}^1F_5^{eoe} + \omega_2^5 \cdot {}^1F_5^{eoo})) \\
 &\quad + \omega_8^5 \cdot (({}^1F_5^{oee} + \omega_2^5 \cdot {}^1F_5^{oeo}) + \omega_4^5 \cdot ({}^1F_5^{ooe} + \omega_2^5 \cdot {}^1F_5^{ooo})) \\
 &= ((f_{eee} + \omega_2^5 \cdot f_{eeo}) + \omega_4^5 \cdot (f_{eoe} + \omega_2^5 \cdot f_{eoo})) \\
 &\quad + \omega_8^5 \cdot ((f_{oee} + \omega_2^5 \cdot f_{oeo}) + \omega_4^5 \cdot (f_{ooe} + \omega_2^5 \cdot f_{ooo})) \\
 &= ((f_{000} + \omega_2^5 \cdot f_{100}) + \omega_4^5 \cdot (f_{010} + \omega_2^5 \cdot f_{110})) \\
 &\quad + \omega_8^5 \cdot ((f_{001} + \omega_2^5 \cdot f_{101}) + \omega_4^5 \cdot (f_{011} + \omega_2^5 \cdot f_{111})) \\
 &= ((f_0 + \omega_2^5 \cdot f_4) + \omega_4^5 \cdot (f_2 + \omega_2^5 \cdot f_6)) \\
 &\quad + \omega_8^5 \cdot ((f_1 + \omega_2^5 \cdot f_5) + \omega_4^5 \cdot (f_3 + \omega_2^5 \cdot f_7))
 \end{aligned}$$

4 Algoritmy a implementace FT

V této kapitole budou prezentovány matlabovské implementace vybraných algoritmů FFT.

4.1 Algoritmus DFT

```
function [F]=four(n)
    N=2^n;
    w=exp(2i*pi/N);
    for i=1:N
        for j=1:N
            F(i,j)=w^((i-1)*(j-1));
        end
    end
end
```

[7]

4.2 Algoritmus dopředné FFT

```
function X=myfft(x);
    N=length(x); n=log2(N);
    % Vytvoreni permutacniho vektoru
    pt=(1:N)';
    i_odd=(1:N/2)*2-1; % liche koeficienty
    i_even=(1:N/2)*2; % sude
    for i=1:n-1
        m=2^(i-1); K=N/(2*m);
        pt_odd=reshape(pt(i_odd),K,m)
        pt_even=reshape(pt(i_even),K,m)
        pt=reshape([pt_odd;pt_even],N,1)
    end;
    % Prenasobovani dilcimi maticemi
    X=x(pt);
    for i=1:n
        K=2^i; m=N/K; % m je pocet bloku B na diag., K je rad B
        w=exp(2i*pi/K);
        D=sqrt(2/K)*diag(w.^(0:-1:-(K/2-1))); %Diag. matice s prvky w
        I=eye(K/2); % Jednotkova matice
        B=[I,D];[I,-D]; % Blok B
        V=diags(B,m); % Matice sestavena z m bloku B
        X=V*X;
    end;
```

[7]

4.3 Algoritmus zpětné FFT

```
function x=myifft(X);
    N=length(X);
    n=log2(N); % n...pocet urovni
    % Vytvoreni permutacniho vektoru pt
    pt=(1:N)';
    i_odd=(1:N/2)*2-1; % i_odd..liche indexy
    i_even=(1:N/2)*2; % i_even.. sude indexy
    for i=1:n-1
        m=2^(i-1);
        K=N/(2*m);
        pt_odd=reshape(pt(i_odd),K,m); % preusporadani prvku
        pt_even=reshape(pt(i_even),K,m);
        pt=reshape([pt_odd;pt_even],N,1);
    end;
    % Prenasobovani dilycimi maticemi
    x=X;
    for i=n:-1:1
        K=2^i;
        m=N/K; % m je pocet bloku B na diagonale, K je rad B
        w=exp(2i*pi/K);
        D=sqrt(2/K)*diag(w.^(0:-1:-(K/2-1))); % Diag. matice s prvky w
        I=eye(K/2); % Jednotkova matice
        B=[[I,D];[I,-D]]; % Blok B
        V=diags(B,m); % Matice sestavena z m bloku B
        x=V'*x;
    end;
    x=x(pt);
```

[7]

4.4 Algoritmus pomocí Daniel-Lanczosovy metody

```

function f=mojeFFT(x)
fm=100; %komplexni frekvence FFT 100Hz
fs=1000; %vykresleni maximalni frekvence na grafu
Ts=1/fs; %iteracni krok
nn=1024; %delka vzorku
isign=1; %pomyslna jednicka

t=0:Ts:(nn-1)*Ts; %vytvoreni casoveho prubehu

%%%% Motylkovy algoritmus%%%%
n=2*nn; %prevedeni vstupniho signalu do 2*nn realneho signalu
j=1;
for i=1:2:n-1
    if(j>i)
        temp=x(j); %prohazovani prvku
        x(j)=x(i);
        x(i)=temp;

        temp=x(j+1);
        x(j+1)=x(i+1);
        x(i+1)=temp;
    end
    m=nn;
    while (m >= 2 && j > m)
        j=j-m; %zmensovani poctu kroku
        m=m/2;
    end
    j=j+m;
end

%%%% DANIEL-Lanczosova metoda %%%%
mmax=2;
while (n > mmax)
    istep=2*mmax;
    theta=isign*(2*pi/mmax);
    wtemp=sin(0.5*theta);
    wpr=-2*wtemp*wtemp;
    wpi=sin(theta);
    wr=1.0;
    wi=0.0;

```

```

for m=1:2:mmax-1
    for i=m:istep:n
        j=i+mmax;
        tempr=wr*x(j)-wi*x(j+1);
        tempi=wr*x(j+1)+wi*x(j);
        x(j)=x(i)-tempr;
        x(j+1)=x(i+1)-tempi;
        x(i)=x(i)+tempr;
        x(i+1)=x(i+1)+tempi;
    end
    wtemp=wr;
    wr=wr*wpr-wi*wpi+wr;
    wi=wi*wpr+wtemp*wpi+wi;
end
mmax=istep;
end
f=x(2:2:n)+1i*x(1:2:n-1);
%plot((0:fs/(nn-1):fs),abs(f))

```

[5]

4.5 Paralelní FFT

Implementovat paralelně FFT je docela složitá úloha a je třeba pracovat se sekvencí N bodů jako s maticí, která má P řádků a N/P sloupců. Nejdříve musí každý procesor paralelně provést přeskládání posloupnosti reverzních bitů (tj. jeden řádek obsluhuje jeden procesor). Poté každý procesor provede N/P bodovou sekvenční FFT na řádcích matice, z důvodu rychlosti nejlépe v paměti cache, stále nekomunikuje s ostatními procesory. Nyní se provede globální transpozice matice (v tomto kroku procesory společně komunikují). Nyní se matice vynásobí otáčecími činiteli. Poté každý procesor provede poslední $\log P$ stupňů výpočtu, tzn. na každém sloupci se provede P -bodové sekvenční FFT, každý procesor obsluhuje N/P^2 sloupců. V posledním kroku se provede globální transpozice matice.

5 Kepstrální analýza

5.1 Historie kepstrální analýzy

V roce 1963 definoval B.P. Bogert *power kepsrum* jako kvadrát absolutní hodnoty FT logaritmu kvadrátu absolutní hodnoty velikosti FT signálu

$$Powerkepsrum = |F \{ \log(|F \{ f(t) \}|^2) \}|^2.$$

Short time kepsrum navrhli vědci Schroeder a Noll pro rozpoznávání lidské řeči. Pro pevně zvolený čas n je STFT (DFT) signálu $x_n[m] = x[m]w[n - m]$. STFT je funkce dvou proměnných, kde n je index, který označuje polohu v Hammingově okně w , které se používá jako typická okenní funkce. Při pohledu na lineární filtr impulsní charakteristiky, má dolní propust frekvenční odezvu, kde mezní kmitočet se mění nepřímo úměrně s délkou okna. To znamená, že pro pevné hodnoty w , $X_n(e^{iw})$ se hodnoty n mění velmi pomalu. STFT je definována jako

$$X_n(e^{iw}) = \sum_{m=-\infty}^{\infty} x[m]w[n - m]e^{-iwm}$$

a STFT kepsrum je definováno jako

$$c_n[m] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log |X_n(e^{iw})| e^{iwm} dw.$$

Komplexní kepsrum definoval Oppenheim ve své práci, která se zabývala teorií homomorfických systémů. Komplexní kepsrum je definováno FT logaritmu FT signálu. V některých literaturách se můžeme setkat s pojmem spektrum spektra

$$K_{ss}(x) = F \{ \log(F(x)) + 2\pi im \}.$$

Reálné používá logaritmus absolutních hodnot FFT. Reálné kepsrum využívá vztahu s power kepstrem. Pokud reálné kepsrum označíme $Rs(x)$ pak vztah je následující

$$Powerkepsrum = (4 * Rs(x))^2.$$

Reálné kepsrum také souvisí s komplexním kepstrem a můžeme jej vyjádřit jako

$$Rs(x) = 0,5 * (KomplexniKepsrum + DobaOtoceniKomplexnihoKepstra).$$

Komplexní kepsrum využívá komplexního logaritmu. **Fázové kepsrum** se dá vyjádřit z komplexního kepsru fáze jako

$$FazoveKepsrum = (KomplexniKepsrum - DobaOtoceniKomplexnihoKepstra)^2.$$

Komplexní kepsrum obsahuje informace o velikosti a fázi počátečního spektra, což nám umožňuje rekonstrukci signálu. Reálné kepsrum využívá pouze velikosti spektra.

V mnoho publikacích můžeme nalézt jednoduchou definici kepsru jako $FFT \rightarrow abs() \rightarrow \log \rightarrow \log \rightarrow IFT$.

5.2 Aplikace kepstra

Na kepstrum se můžeme dívat pohledem, že je to rychlost změn v různých pásmech spektra. Původní využití bylo charakterizovat ozvěny, které vznikaly z výbuchu bomb a zemětřesení. Také se využívá k určení základní frekvence lidské řeči a dokážeme pomocí spektra analyzovat radarové vysílání. Pitch metoda je velmi efektivní, protože zvukové vibrace a zvukové roložené kepstra jsou pomocí logaritmu power kepstra zřetelně odděleny. Kepstrum využívá homomorfní zpracování signálu, kde převáděný signál je zpracován kombinací konvolucí a lineární separací. *Power kepstrum* je velmi hodně využíváno na analýzu lidské řeči. V tomto případě se nejdříve spektrum transformuje do tzv. MFC stupnice. MFC je „mel-frequency“, která se používá pro detekci základního tónu, hlasové identifikace a dalších věcí. Kepstrum je užitečné z tohoto důvodu, protože nízkofrekvenční periodické buzení, které vychází z hlasivek se dá vyčistit filtrací vokálního traktu, které se šíří v časové oblasti a jeho frekvence se neustále mění, avšak v některých oblastech jsou periodické. [10]

5.3 Kepstrální pojmy

Nezávislé proměnné kepstrálního grafu se nazývají „quefreny“. Quefreny je míra času. Jako příklad vezměme vzorkovací frekvenci audio signálu 53200 Hz a frekvenci spektra, která bude 100 vzorků, pak výsledná frekvence pitch metody bude $53200/100=532$ Hz. Tento vrchol se vyskytuje v kepstru několikrát, protože signál je harmonický. Čistý sinusový signál by se nikdy neměl používat k testování spektra, protože žádný signál nemá tak čisté harmonické opakování jako funkce sinus. Na testovací signál bychom měli spíše aplikovat součet více funkcí sinus s rozdílnými argumenty [10].

5.4 Liftrování kepstra

Pokud chceme filtrovat kepstrum, tato operace se nazývá liftrování. Dolní propust liftrování je stejná jako dolní propust filtrování kepstra ve frekvenčním spektru frekvenční oblasti. Liftrování se implementuje pomocí násobením oknem v „quefreny“ oblasti a poté ihned převedeme kepstrum do frekvenční oblasti, což pak ve výsledku má za následek nižší hladinu signálu. [10]

5.5 Konvoluce signálů

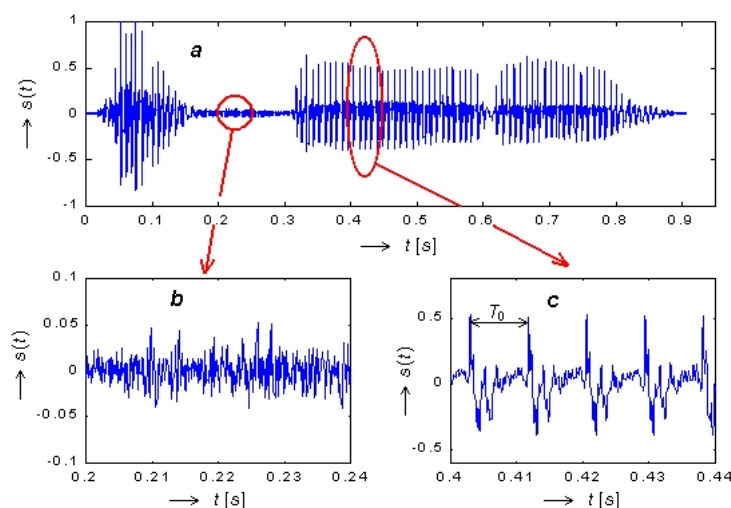
Velmi důležitá vlastnost kepstra je ta, že kepstrum konvoluce dvou vstupních signálů může být vyjádřena jejich komplexními kepstr [10]

$$x_1 * x_2 \rightarrow x'_1 x'_2.$$

5.6 Charakter řečového signálu

Lidská řeč je spojitá s časově proměnlivým procesem. Přenáší informaci od řečníka k posluchači a je zakódována a přenášena pomocí akustického vlnění. Řeč se vytváří

Obrázek 5: Rozdělení signálu na znělou a neznělou část [4]



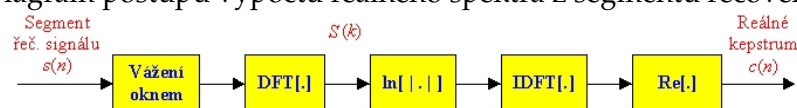
ovlivňováním výdechu proudu vzduchu z plic hlasovým ústrojím člověka, začínajícího hlasivkami a končícího rty. Řečovým signálem rozumíme posloupnost diskrétních vzorků signálu, který můžeme obdržet z mikrofону. Pokud tento signál podrobíme analýze, můžeme si všimnout, že některé oblasti jsou víceméně periodické, v tomto případě jde o znělé části, periodu označíme symbolem T_0 . Potom můžeme nalézt také oblasti, které mají charakter šumu, tyto oblasti jsou neznělé části řeči. Znělost a neznělost je způsobená tím, jestli výdechový proud vzduchu z plic rozkmitá hlasivkovou štěrbinu nebo ne.

Na obrázku 5 si můžeme prohlédnout průběh řečového signálu při promluvení slova „aféra“, kde obrázek b je detailem neznělého úseku a obrázek c je detail znělého úseku.[8]

5.7 Model vytváření řeči

Základním modelem hlasového traktu je ten, který pomocí co nejmenšího počtu parametrů $c(n)$ napodobuje přenosové vlastnosti lidského hlasového traktu. Hlasový trakt člověka je tvořen soustavou dutin, tj. dutina hrdelní, ústní, nosní, do této definice se zahrnují i zuby, jazyk a rty [8]. Tento model se dá nasimulovat kupříkladu pomocí číslicového filtru. Při mluvení se velikost dutin spojitě mění, tzn. mění se i jeho přenosové vlastnosti. Je třeba, aby byl filtr, který modeluje hlasový trakt časově nestacionární. Jelikož změna rozměrů hlasového traktu je relativně malá, můžeme se dopustit malé odchylky, jestliže budeme počítat, že na intervalu 10 až 30 ms se vůbec odchylka neprojeví. Filtr modelující hlasový trakt může mít uvnitř časového intervalu konstantní parametry $c(n)$, které budou pro každý segment jiné. Typ buzení hlasového traktu je závislý na tom, pokud je segment řeči znělý nebo neznělý. Pokud je řeč znělá, modelují se kmitání hlasivek generátorem pulsů, kde perioda odpovídá základní hlasivkové periodě T_0 . V neznělé části řeči je možné vytvářet buzení modelu hlasového traktu generátorem bílého šumu s plochým výkonovým spektrem. V reálných případech není buzení vždy jen znělé nebo

Obrázek 6: Diagram postupu výpočtu reálného spektra z segmentu řečového signálu [4]



neznělé. Pro větší přiblížení skutečnosti se využívá určitého směřování těchto zdrojů a nebo také pokud se jedná o znělé buzení se přimíchává k pulsům generátoru barevný šum s definovaným výkonovým spektrem.

5.8 Reálné kepstrum

Definice: Necht' $S(k)$ je diskrétní spektrum analyzovaného signálu, které může být určeno pomocí DFT

$$S(k) = DFT \{s(n)\} = \sum_{n=0}^{N-1} s(n) e^{-i \frac{2\pi}{N} kn},$$

kde k označuje frekvenční index, který nabývá diskrétních hodnot $0, 1, \dots, N$. N je počet uzlů v algoritmu FFT [8]. Ke každému takovému indexu náleží frekvence $f = \frac{k}{N} f_{vz}$. Reálné kepstrum poté dostaneme jako reálnou část zpětné DFT logaritmu modulu spektrální funkce $S(k)$

$$c(n) = Re \{ DFT^{-1} [\log |S(k)|] \} = Re \left\{ \frac{1}{N} \sum_{k=0}^{N-1} \log |S(k)| e^{i \frac{2\pi}{N} kn} \right\}.$$

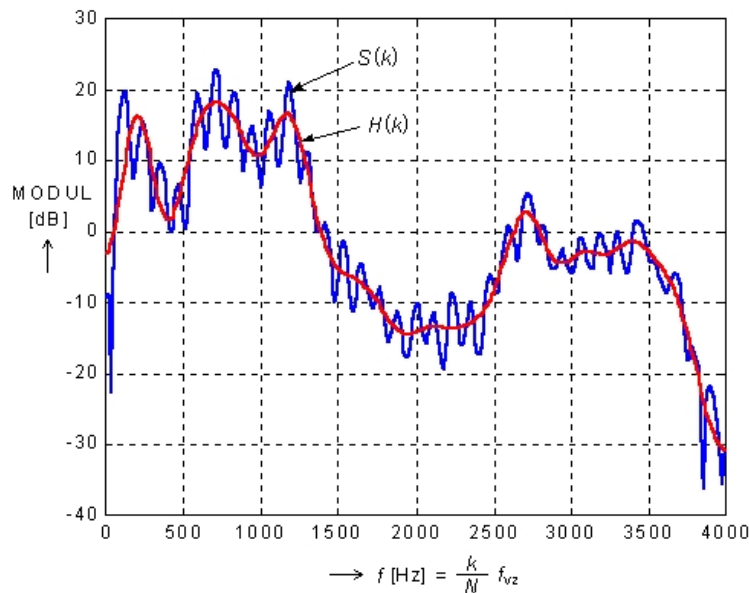
5.9 Výpočet reálného spektra

Jelikož je vstupní signál časově proměnlivý a chceme tuto změnu odchytnout, musíme analyzovat segment takové délky, kde předpokládáme stacionaritu signálu. Existují dvě základní metody segmentování řečového signálu:

- pitch synchronní segmentace - délka segmentu odpovídá jedné periodě znělé řeči
- pitch asynchronní segmentace - délka segmentu je pevná a nezávislá na základní periodě řeči, většinou je rovna dvěma až třem pitch periodám.

První metoda není moc využívána, protože je jak implementací tak výpočetně mnohem náročnější a touto metodou se ztratí informace o základním tónu kepstra. Při určování pitch asynchronní segmentace se vždy dopouštíme nějaké chyby z důvodu toho, že segment neodpovídá přesné periodě signálu. V nejlepším případě by bylo nejlepší použít délku periody signálu, ale toto je není možné, protože bychom porušovali stacionaritu řečového signálu uvnitř segmentu. Jako kompromis se používá kombinace délky segmentu a přesnosti spektra. Aby byla analýza co nejpřesnější používá se překrývání segmentů a každý segment se musí vynásobit váhovací posloupností. Při analýze řeči se nejčastěji využívá Hammingovo okno. Z vlastností FT plyne, že konvoluce dvou signálů,

Obrázek 7: Diskrétní spektrum samohlásky „a“ [4]



tj. $x(n)$ a $h(n)$ se převede na součin FT

$$DFT\{s(n)\} = DFT\{x(n) * h(n)\} = DFT\{x(n)\} DFT\{h(n)\} = X(k)H(k) = S(k).$$

Jestliže na tento vztah aplikujeme přirozený logaritmus, využijeme pravidla, že logaritmus součinu je součet logaritmů jednotlivých činitelů

$$\log [|S(k)|] = \log [|X(k)H(k)|] = \log [|X(k)|] + \log [|H(k)|].$$

Na tomto obrázku vidíme diskretní spektrum stacionárního úseku samohlásky „a“, které je symbolizováno modrou barvou. Červená křivka označuje pomalou změnu průběhu spektra, kterou určuje přenos hlasového traktu. Těmito operacemi se převedla konvoluce v časové oblasti na součet logaritmů diskretních modulových spekter. V tomto případě můžeme použít zpětnou FT na součet signálů. Jestliže budeme zobrazovat jen reálné složky, dostaneme reálné kepstrum

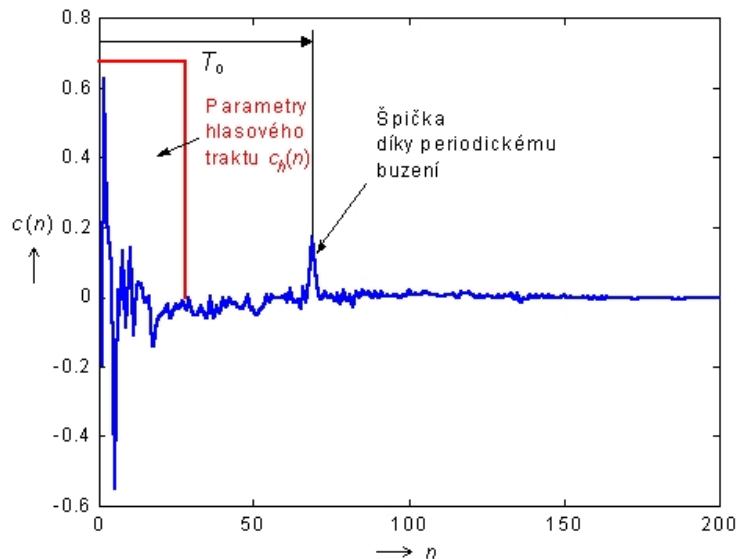
$$c_x(n) = \text{Re} \{ DFT^{-1} [\ln |X(k)|] \}$$

$$c_h(n) = \text{Re} \{ DFT^{-1} [\ln |H(k)|] \}$$

$$c(n) = \text{Re} \{ DFT^{-1} [\ln |S(k)|] \} = c_x(n) + c_h(n).$$

Složky kepstra $c_x(n)$ jsou tedy ty, které nebudou jako parametry hlasového traktu, jsou vytvářeny reálným budícím signálem $x(n)$. Většinou se však zanedbávají parametry budícího signálu. Jestliže budeme vyšetřovat spektrum pouze o jedné periodě znělého úseku

Obrázek 8: Reálné kepstrum stracionárního segmentu znělé hlásky „a“ [4]



(při pitch synchronní segmentaci), ztratí se informace o buzení. V tomto případě modulové kepstrum nebude zvlněné a nebude existovat špička v kepstru. Jestliže se vyšetřuje neznělá část řeči, spektrum obsahuje jak pomalé tak rychle měnící se složky. Pomalé změny se považují za modulové spektrum hlasového traktu, rychlé změny v tomto případě nejsou periodické a chovají se náhodně. Proto se u něznělých částí řeči nevyskytuje špička.

5.10 Homomorfická filtrace

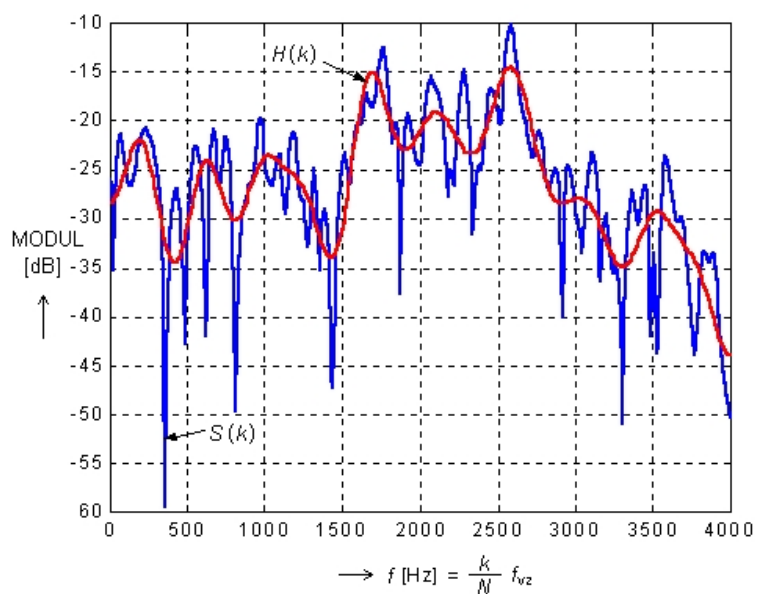
Jestliže chceme mít co nejhladší modulové spektrum hlasového traktu, můžeme využít několik počátečních kepstrálních koeficientů $c_h(n)$, které budeme vybírat metodou pravúhlého okna. Tato operace se nezývá homomorfická filtrace [8]. V této metodě se využívá opačný postup, než který bychom zvolili při výpočtu reálného kepstra ze zadaného modulového spektra. Nyní musíme upravit reálné kepstrální koeficienty $c(n)$ na minimální fázovou část komplexního spektra $k(n)$, protože jsme při výpočtu spočetli pouze modul reálného kepstra

$$k(n) = c(n), n = 0, n = N/2$$

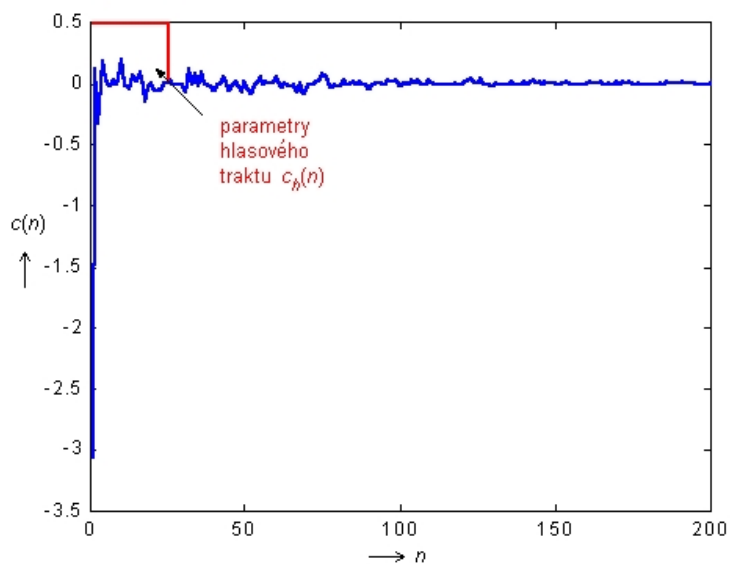
$$k(n) = 2c(n), 1 \leq n \leq N/2$$

$$k(n) = 0, N/2 < n \leq N - 1.$$

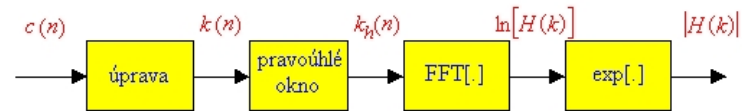
Obrázek 9: Modulové spektrum stacionárního segmentu neznělé hlásky a jeho vyhlazení [4]



Obrázek 10: Reálné kepstrum stacionárního segmentu neznělé hlásky „f“ [4]



Obrázek 11: Schéma výpočtu vyhlazeného modulového spektra metodou homomorfní filtrace [4]



Pro neznělý úsek bude hodnota $c(n)$ výrazně záporná, protože neznělý úsek má výrazně menší amplitudu než znělý. Vyhlazené spektrum získáme jako

$$|H(k)| = \exp \{ DFT[k_h(n)] \} = \exp \left\{ \sum_{n=0}^{N-1} k_h(n) e^{-i \frac{2\pi}{N} kn} \right\}.$$

Abychom optimálně zvolili počet keprstrálních koeficientů, řídíme se vzorkovací frekvencí. Bylo zjištěno, že při $f_{vz} = 8$ kHz je optimální zvolit $N_0 = 26$ keprstrálních koeficientů. Při využití jiné vzorkovací frekvence s určuje N_0 přenásobením $N_0/f_{vz} = 26/8$.

6 Liftering

Liftrování znamená to stejné jako filtrování ve quefreny doméně, kde požadovaná analýza frekvenční části je vybírána pomocí násobení zadaného kepstra obdelníkovým oknem na požadované pozici. Existují dva typy liftrování a to **low-time liftering** a **high-time liftering**. Low-time liftering operace se provádí extrahováním vokálního traktu charakteristiky v quefreny doméně a high-time liftering se provádí rozšíření budící charakteristiky analýzy hlasového rámce.

6.1 Low-time liftering

Low-time liftering se používá pro odhad pomalu měnící se hodnot na hlasové charakteristické oblaci které jsou vypočteny kepstem dané řečové posloupnosti. Low-time liftering okno se používá pro extrakci charakteristiky hlasového ústrojí a může být reprezentována následovně

$$w_e[n] = 1, 0 \leq n \leq L_c$$

$$w_e[n] = 0, L_c \leq n \leq \frac{N}{2},$$

kde L_c je část odstraněné délky liftrovaného okna a $N/2$ je polovina celkové délky kepstra. Obvykle se hodnota L_c nastavuje na 15 nebo 20. Vokální trakt charakteristiky lze získat vynásobením kepstra $c(n)$ s low-time liftering oknem jak je ukázáno na rovnici níže

$$c_e(n) = w_e[n] \cdot c(n).$$

Aplikací DFT a low-time lifteringu vezememe jako logaritmus velikosti spektra, které je hlasovým traktem krátkého záznamu hlasu

$$\log [|H(w)|] = DFT[c_e(n)].$$

Důležitými parametry vokálního traktu myslíme formant umístění a šířku pásma, kterou můžeme spočítat z hlasového spektra. Formantované místa lze odhadnout pomocí vybírání vrcholů z hladkého vokálního traktu spektra. Blokový diagram níže ukazuje, jak postupně se využívá low-time liftrování.

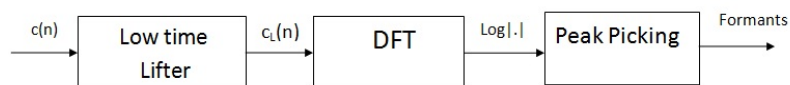
6.2 High-time liftering pro odhad intervalu

Jako kepstrum, které je vypočítáváno z analýzy posloupnosti zvukového záznamu a je symetrické, je polovina délky kepstra považována za liftering. Pokud rozšíříme charakteristiku získáme prostřednictvím high-time lifteringu operaci, která využívá tuto metodu okna

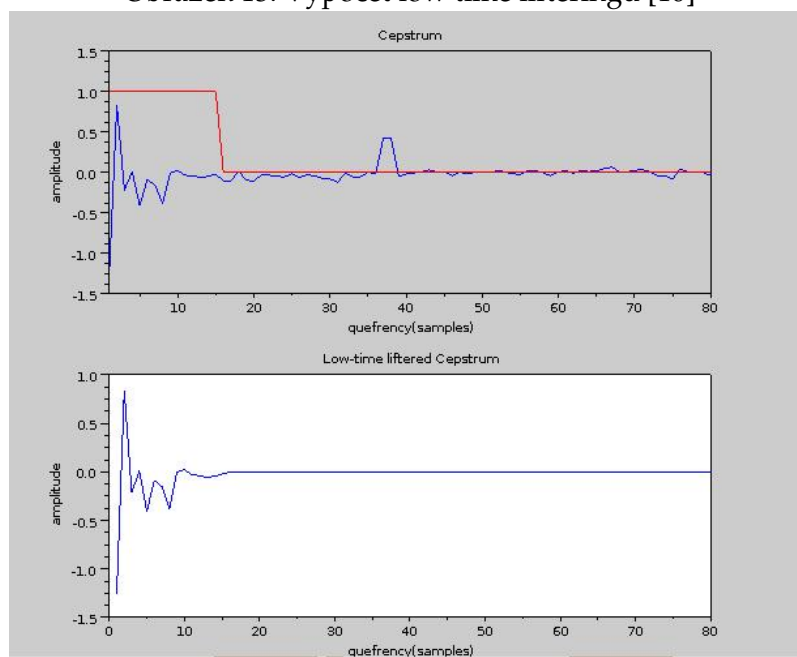
$$w_h[n] = 1, L_c \leq n \leq \frac{N}{2}$$

$$w_h[n] = 0, jinak,$$

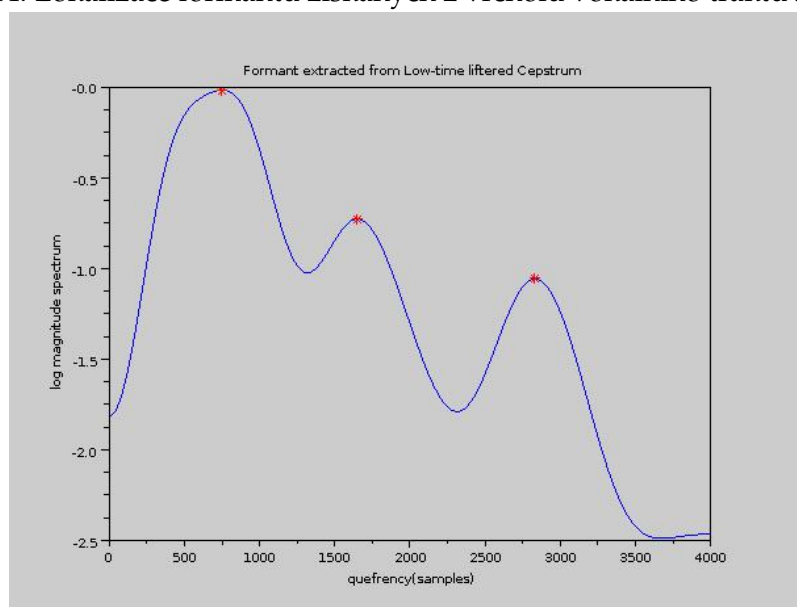
Obrázek 12: Diagram [10]



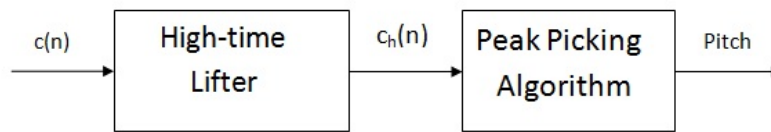
Obrázek 13: Výpočet low time lifteringu [10]



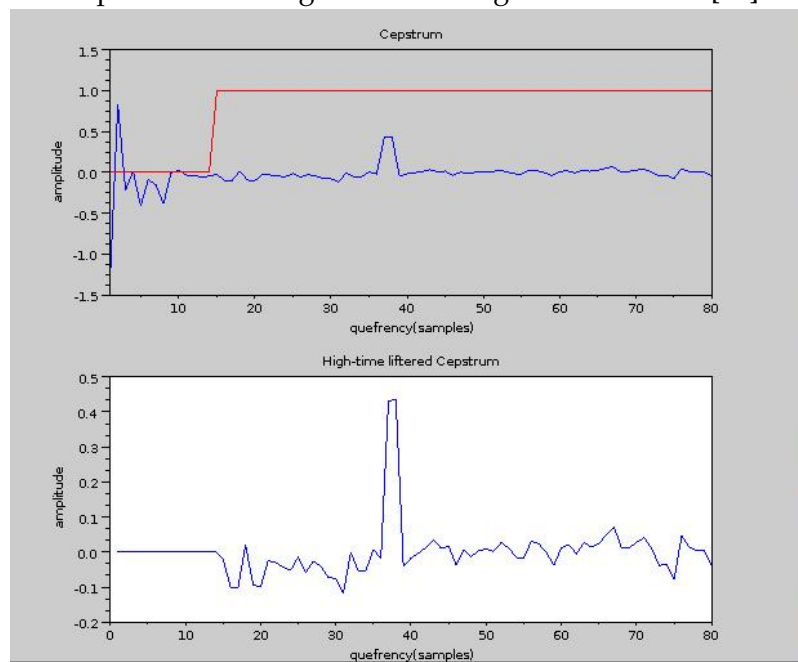
Obrázek 14: Lokalizace formantů získaných z vrcholů vokálního traktu spektra [10]



Obrázek 15: Blokový diagram procesu výpočtu [10]



Obrázek 16: Kepstrum hlasového segmentu a lifterované okno (v červené barvě) a část vokálního traktu kepstra získané high-time liftering metodou oken [10]



kde L_c reprezentuje ořezanou délku lifterovaného okna a $N/2$ je polovina délky kepstra. Obvykle L_c se nastavuje na hodnotu 15 nebo 20. Rozšíření charakteristiky je získáváno násobením high-time liftering okna a kepstra

$$c_h = w_h \cdot c(n).$$

Diagram na obrázku níže ukazuje proces jak získáme odhad intervalu pomocí high-time liftering. V intervalu lze odhadnout okamžik, který odpovídá nejvyššímu vrcholu v high-time liftrovaném kepstru. Interval liftrovaného kepstra vynásobený vzorkovací frekvencí vrací frekvenční rozteč analýzy řeči.

7 Numerické experimenty

7.1 Porovnání rychlosti algoritmů FFT

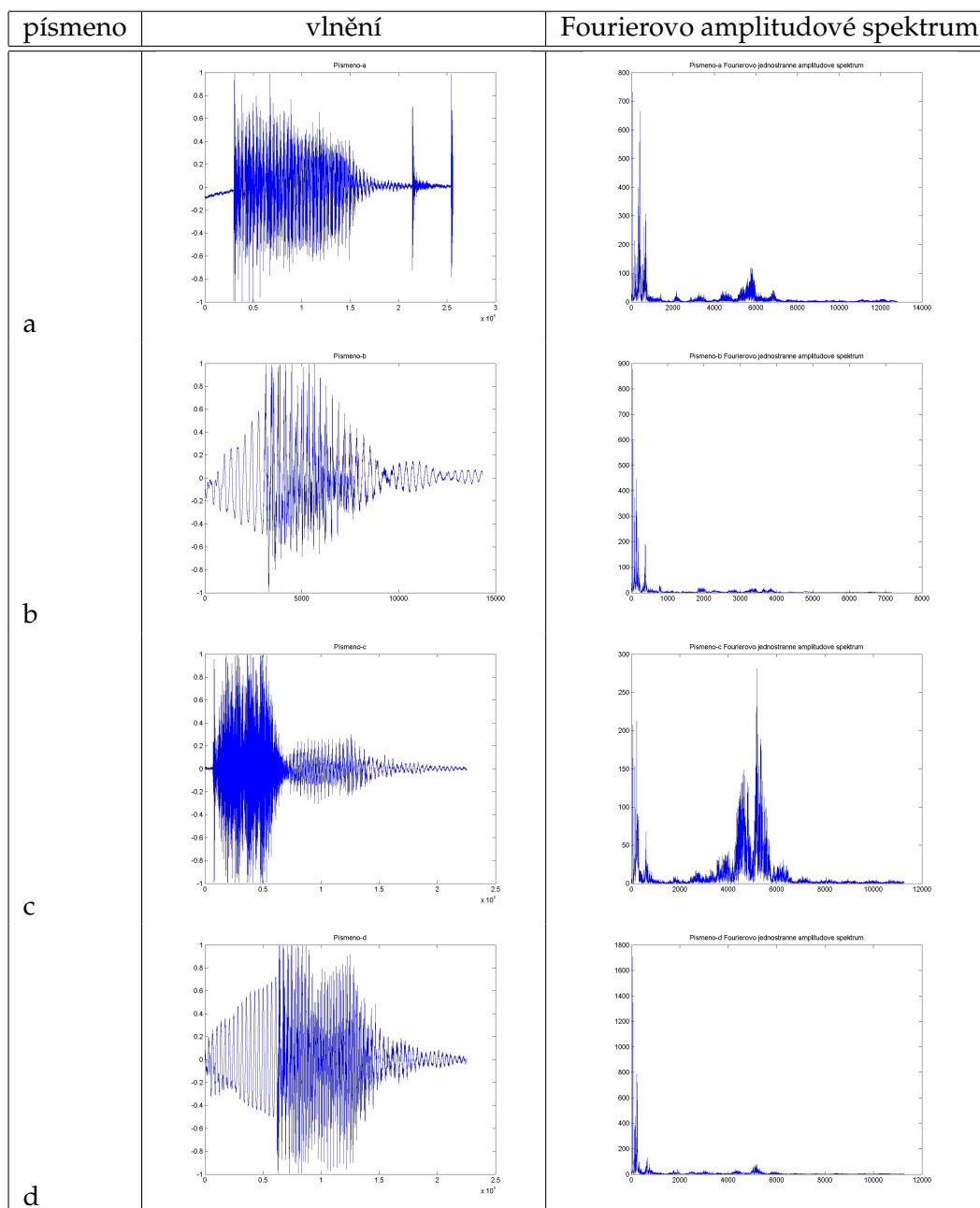
Na vstupu jsme zadali vektor náhodných čísel v intervalu (0,1) o velikosti 2048 prvků. Poté jsme vzali různé algoritmy a ve smyčce o 1000 opakování jsme počítali dobu, jakou trvá provedení jednoho algoritmu. Z těchto výsledků jsme poté vypočítali průměrnou dobu trvání. Nyní vyzkoušíme jak se liší rychlosti algoritmů. Poté jsme provedli stejné měření na vektoru o velikosti 10240.

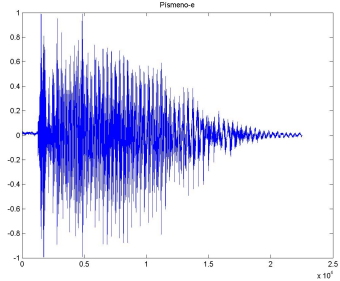
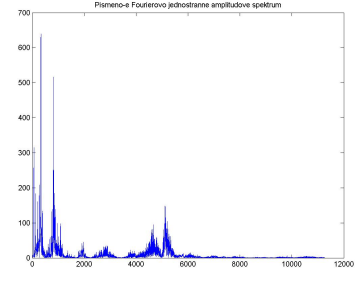
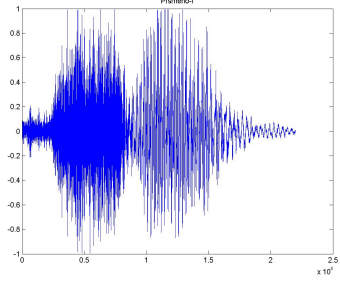
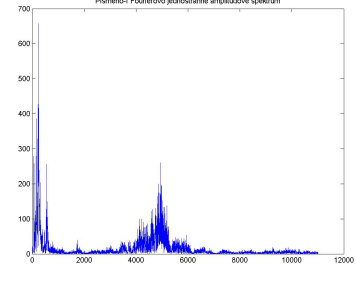
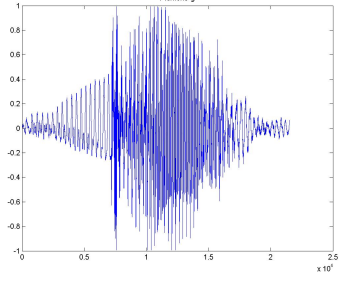
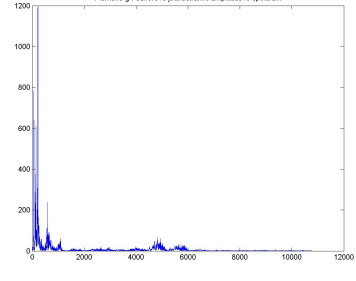
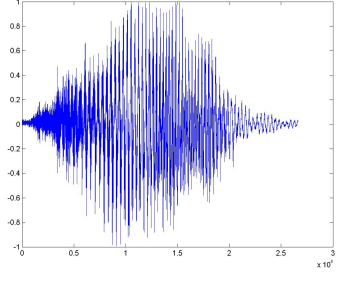
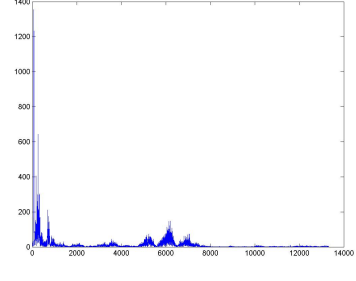
počet prvků	Matlabovská FFT [s]	dopředná FFT [s]	Daniel-Lanczos [s]	DFT [s]
2048	0.000010	0.001160	0.000591	470.358533
10240	0.000210	1261.302	0.009189	-

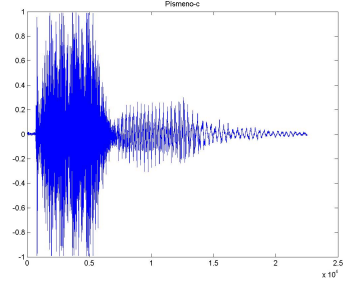
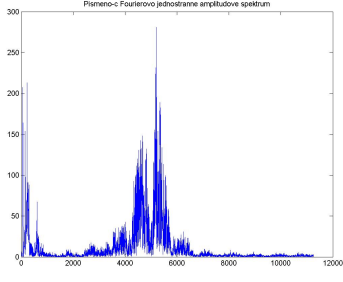
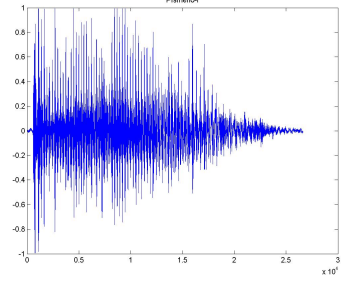
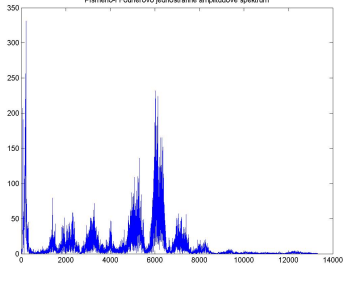
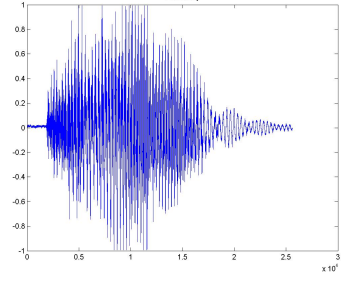
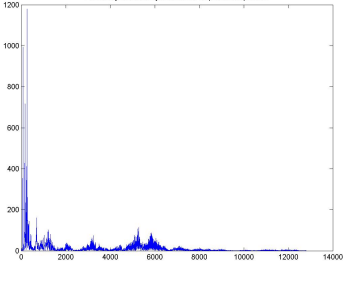
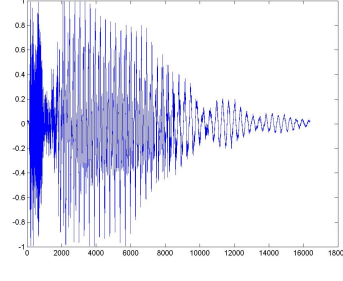
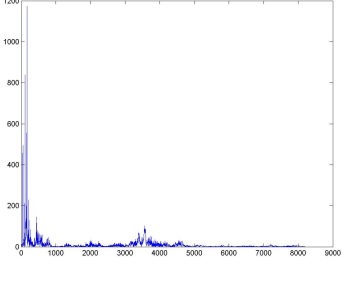
Lze vidět, že bezkonkurenčně nejrychlejší implementace je ta Matlabovská. Také jde vidět, že je o dost vhodnější pro výpočet vlastního FFT použít Daniel-Lanczosovu metodu, která je o 0,01s pro vektor velikosti 2048 rychlejší než původní výpočet pomocí dopředné FFT. Nejméně nevhodné je používat DFT metodu, která je neskutečně pomalá a pro větší vstupní data ji počítač nevypočítá. Pro větší vektory je vhodné využívat pouze Matlabovskou metodu nebo metodu Daniel-Lanczos.

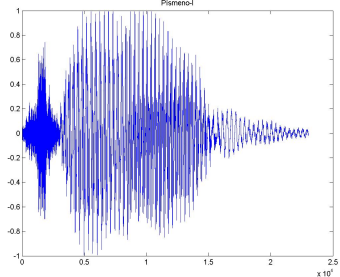
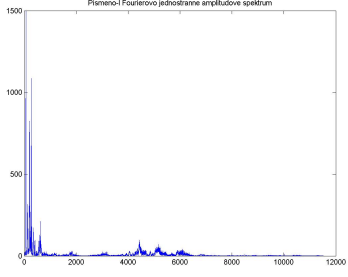
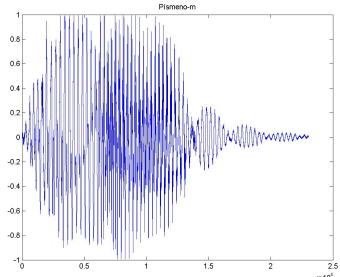
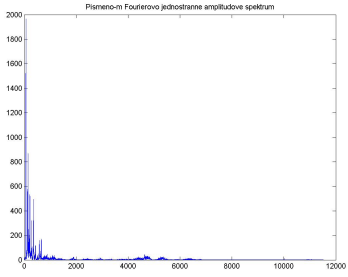
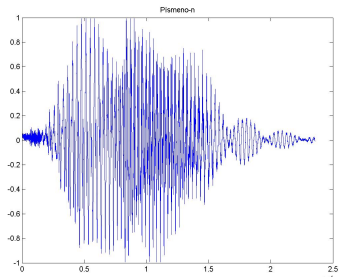
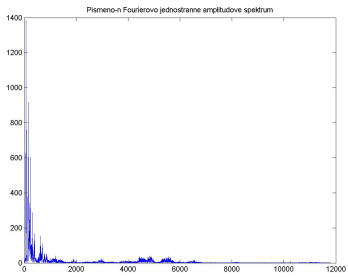
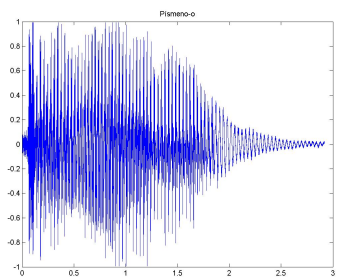
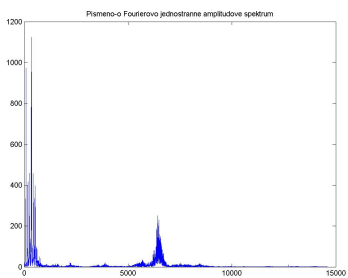
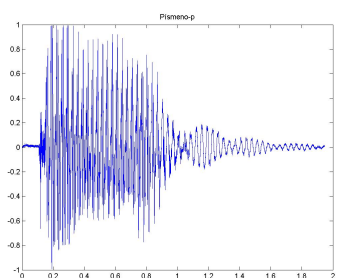
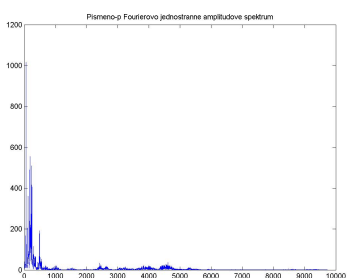
7.2 Spektrální analýza české abecedy

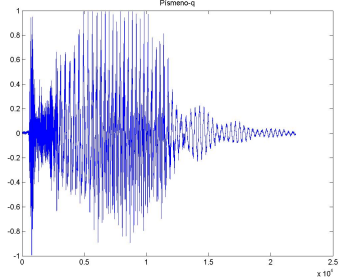
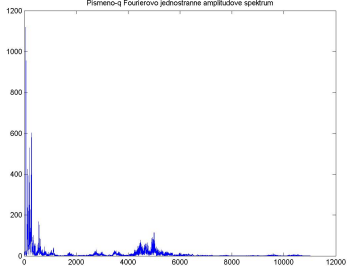
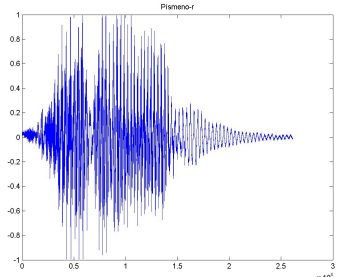
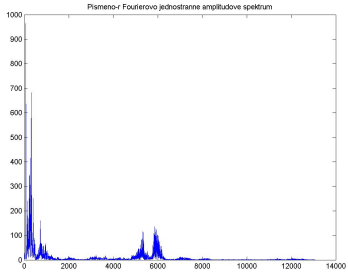
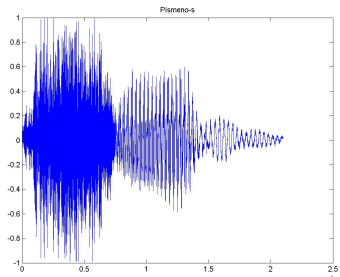
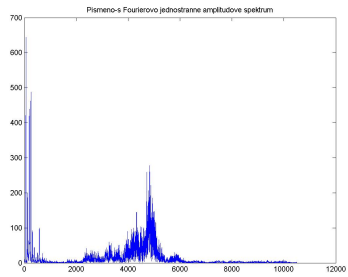
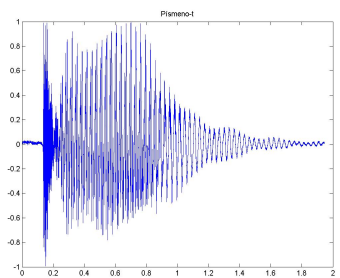
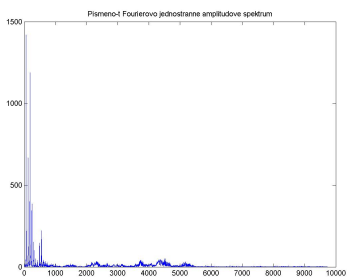
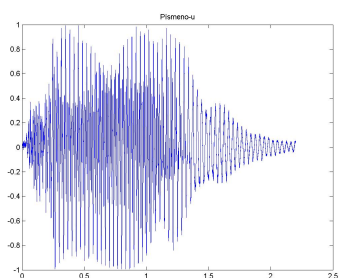
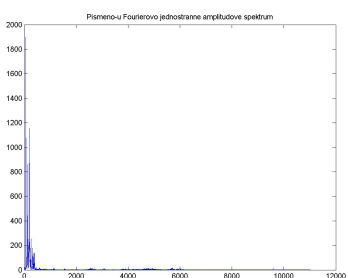
Nyní provedeme spektrální analýzu na české abecedě.

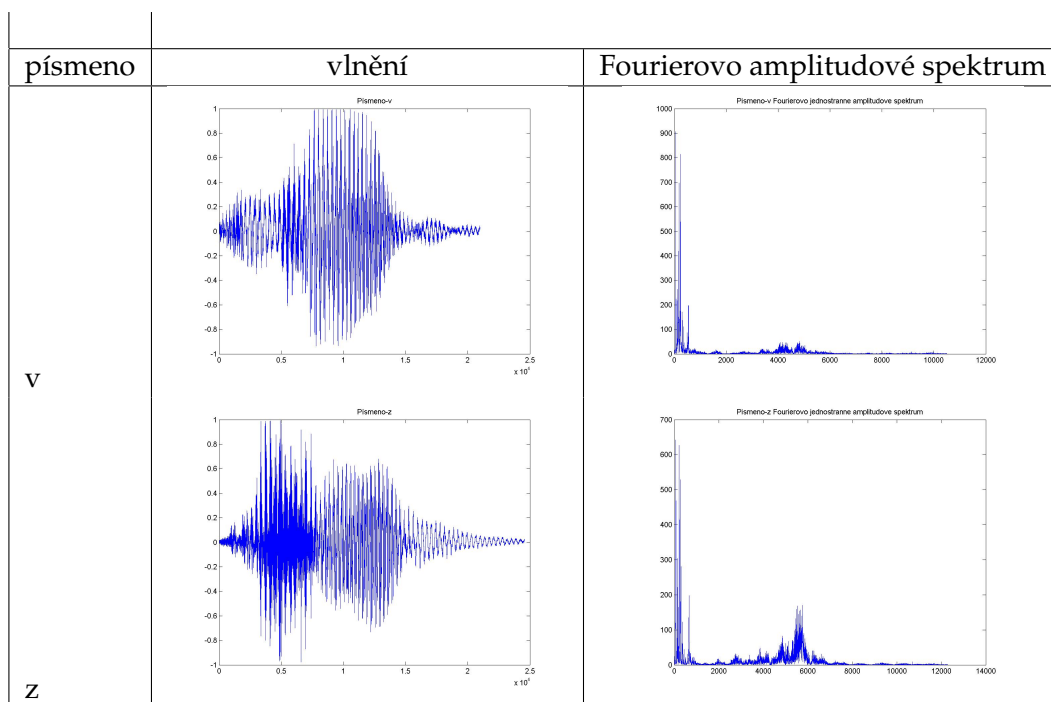


písmeno	vlnění	Fourierovo amplitudové spektrum
e		
f		
g		
h		

písmeno	vlnění	Fourierovo amplitudové spektrum
ch i j k		
		
		
		

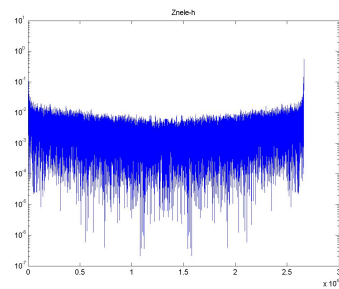
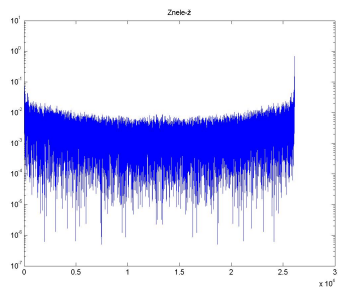
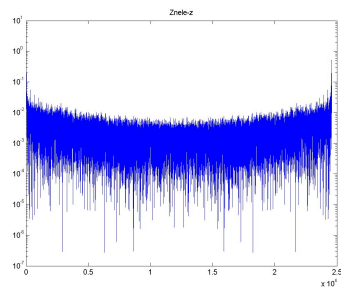
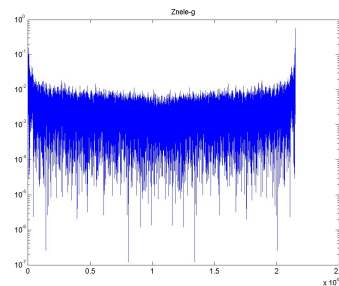
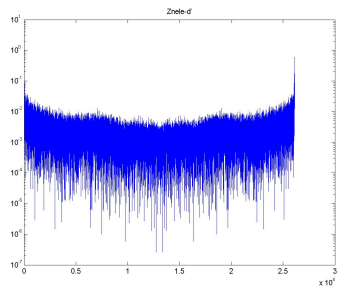
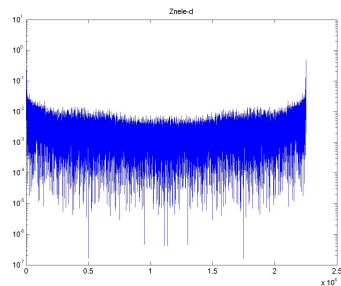
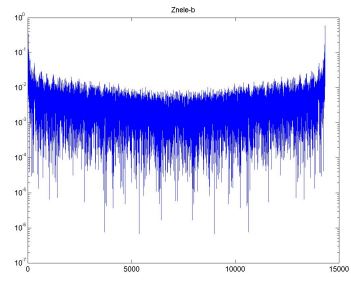
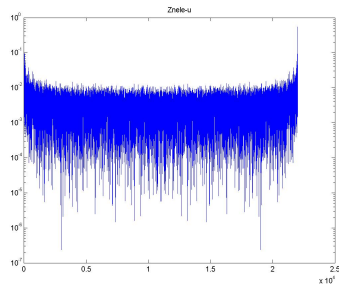
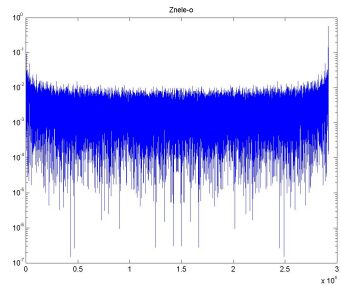
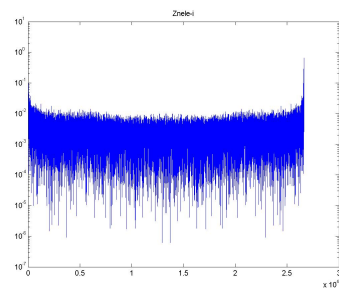
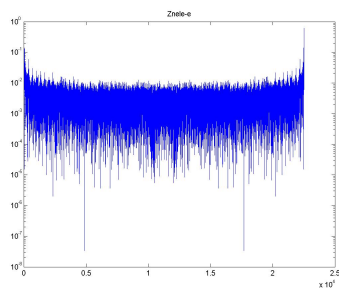
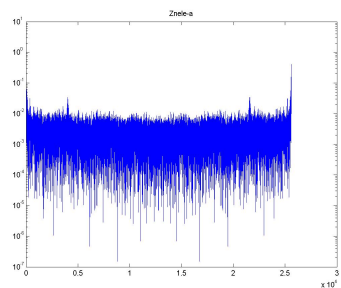
písmeno	vlnění	Fourierovo amplitudové spektrum
l m n o p		
		
		
		
		

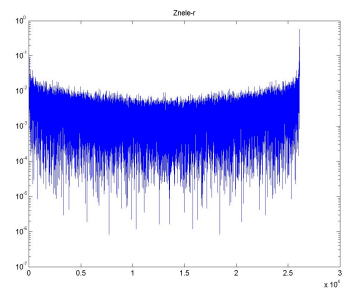
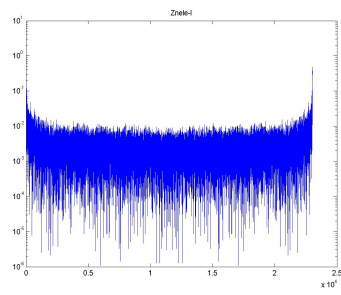
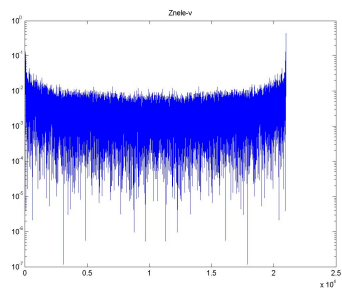
písmeno	vlnění	Fourierovo amplitudové spektrum
q		
		
r		
		
t		
u		



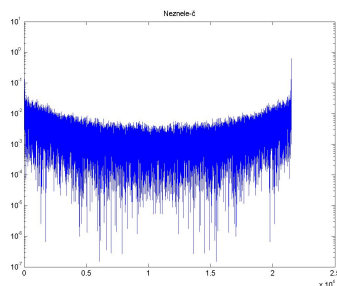
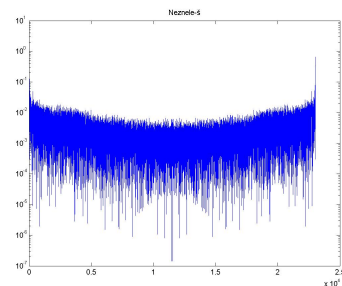
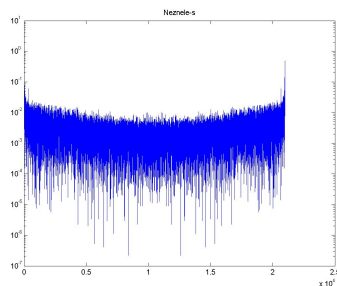
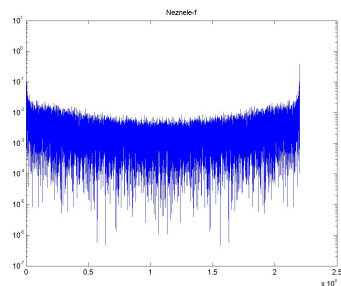
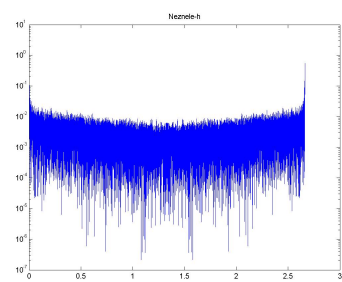
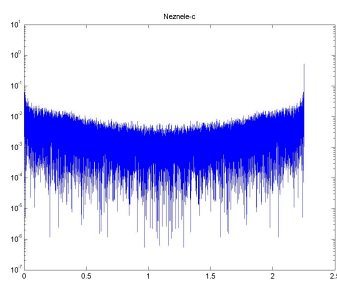
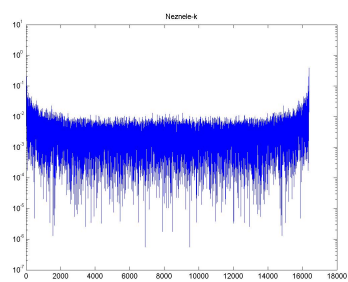
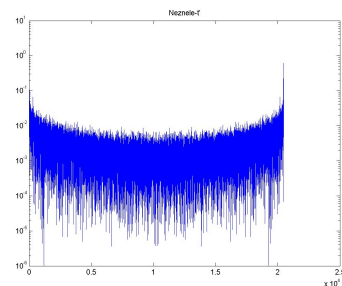
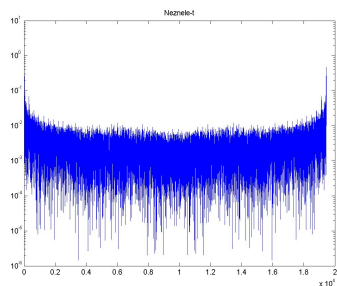
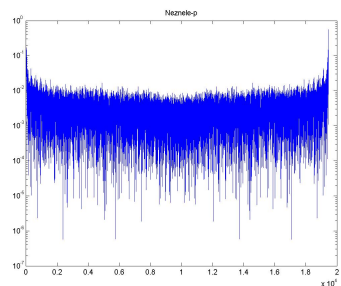
7.3 Reálná kepstra

7.3.1 Znělé hlásky

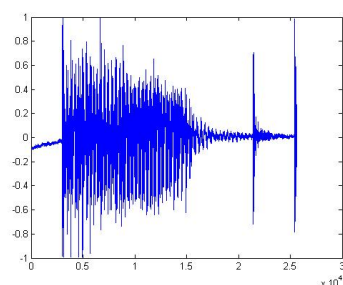
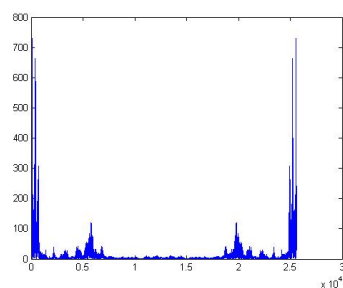
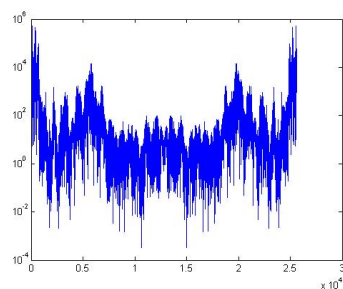




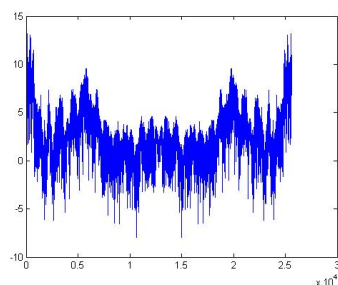
7.3.2 Neznělé hlásky



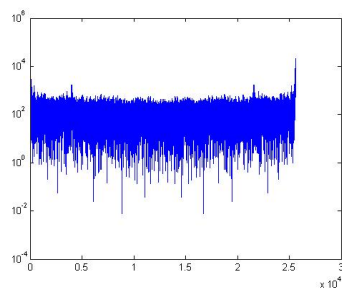
7.4 Analýza písmene „a“

Obrázek 17: Signál zvukového záznamu $z = y(t)$ Obrázek 18: Fourierovo amplitudové spektrum $z = |FFT(y(t))|$ Obrázek 19: Power spektrum $z = |FFT(y(t))|^2$ 

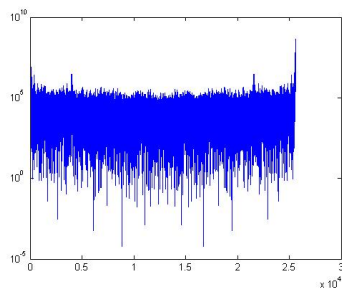
Obrázek 20: Diskrétní spektrum $z = \log |FFT(y(t))|^2$



Obrázek 21: Amplitudové spektrum diskrétního spektra $z = \left| FFT(\log |FFT(y(t))|^2) \right|$

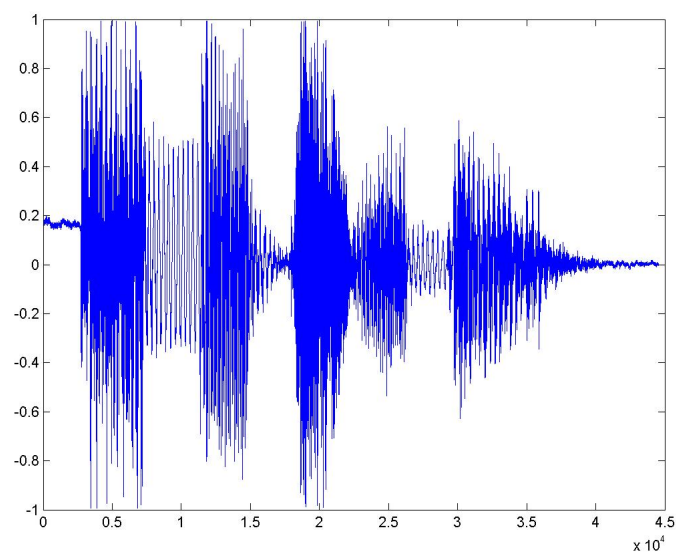


Obrázek 22: Power kepsrum $z = \left| FFT(\log |FFT(y(t))|^2) \right|^2$

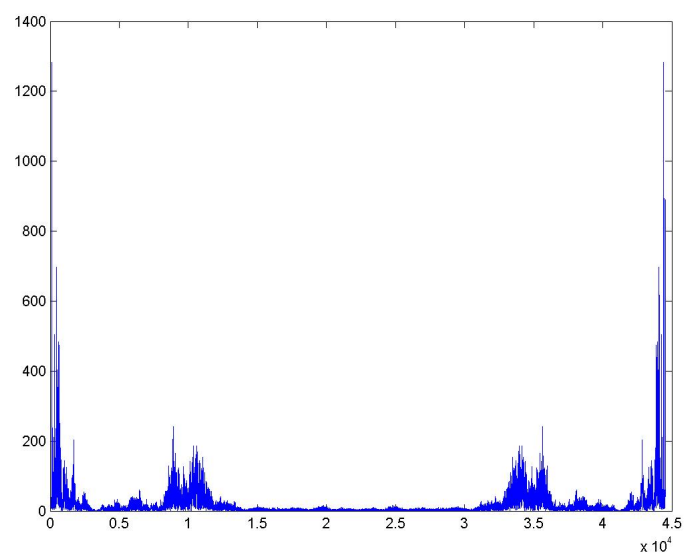


7.5 Analýza slova „abeceda“

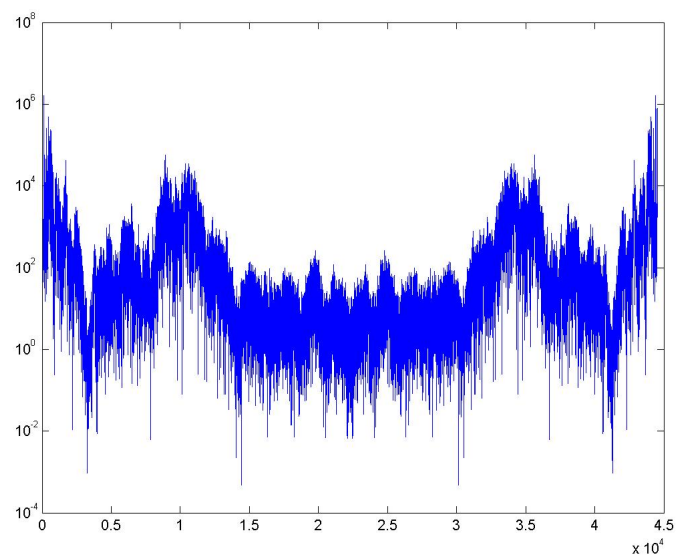
Obrázek 23: Signál zvukového záznamu $z = y(t)$



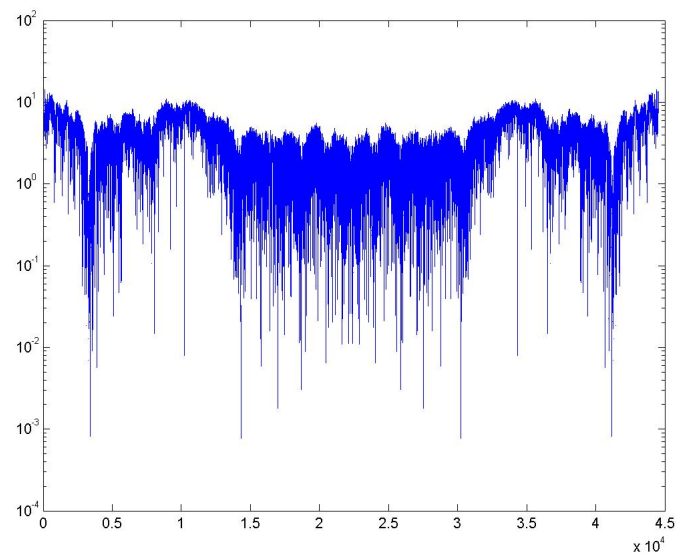
Obrázek 24: Fourierovo amplitudové spektrum $z = |FFT(y(t))|$



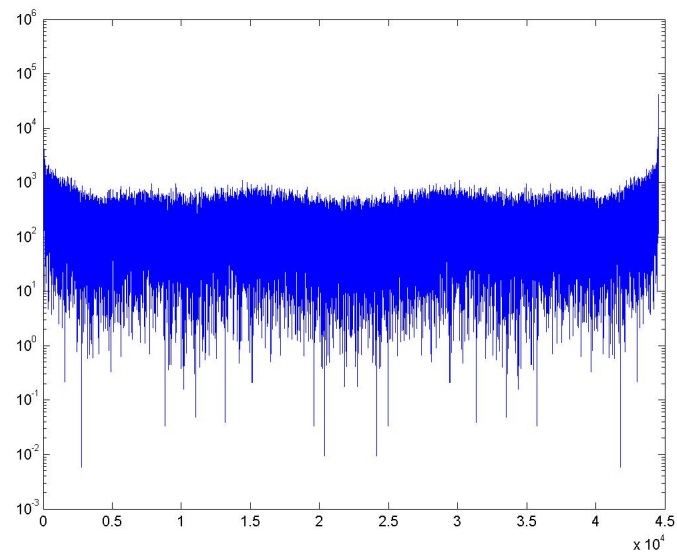
Obrázek 25: Power spektrum $z = |FFT(y(t))|^2$



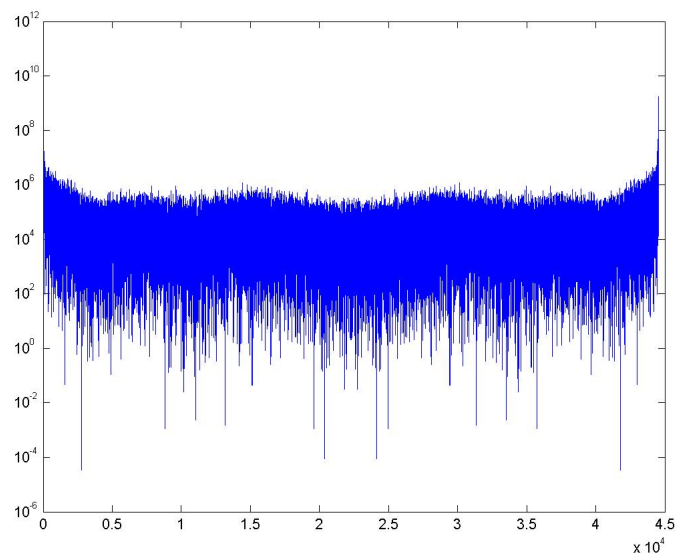
Obrázek 26: Diskrétní spektrum $z = \log |FFT(y(t))|^2$



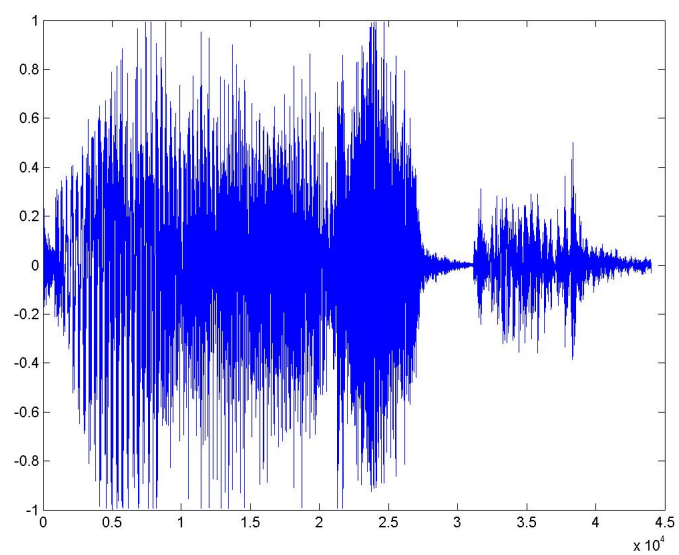
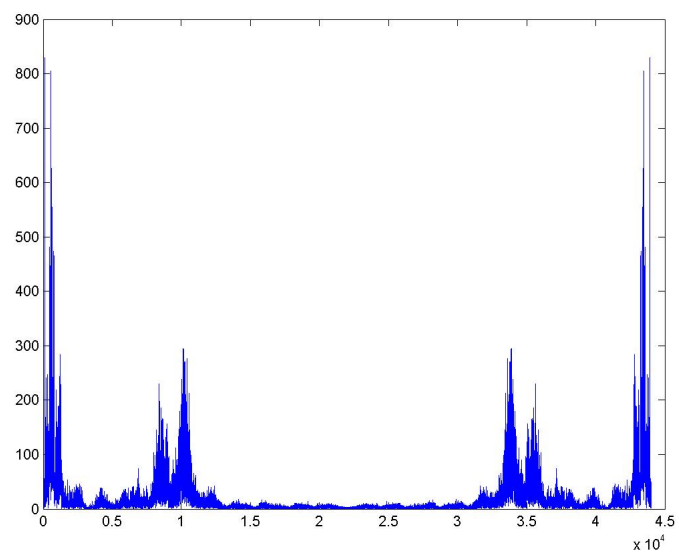
Obrázek 27: Amplitudové spektrum diskrétního spektra $z = \left| FFT(\log |FFT(y(t))|^2) \right|$

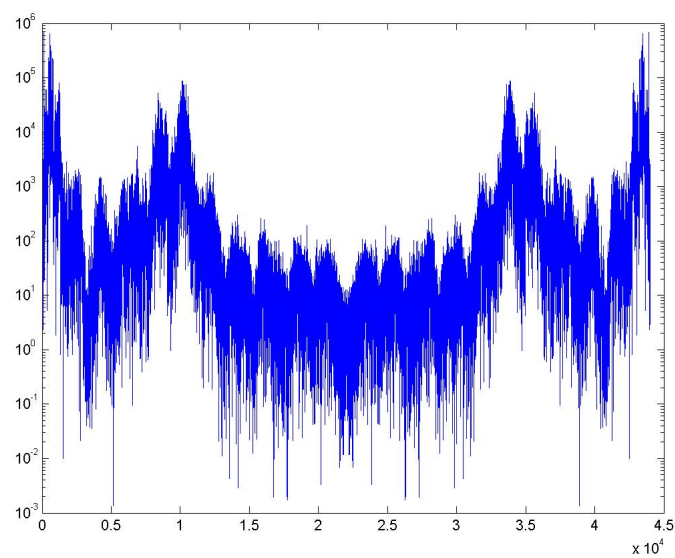
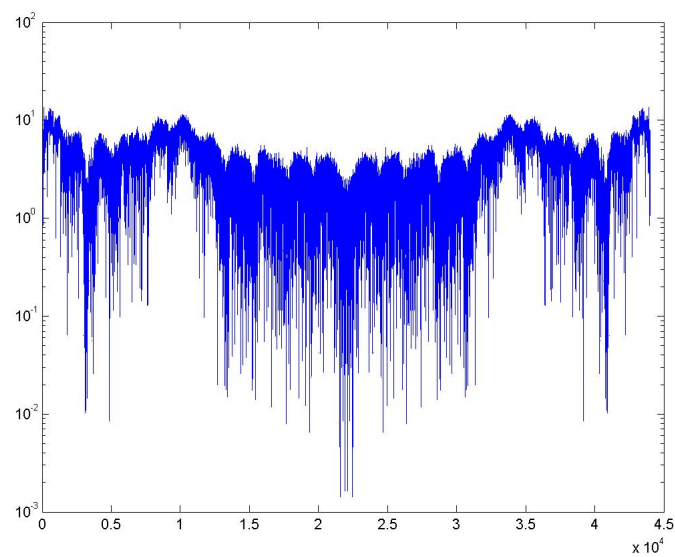


Obrázek 28: Power keprum $z = \left| \text{FFT}(\log |\text{FFT}(y(t))|^2) \right|^2$

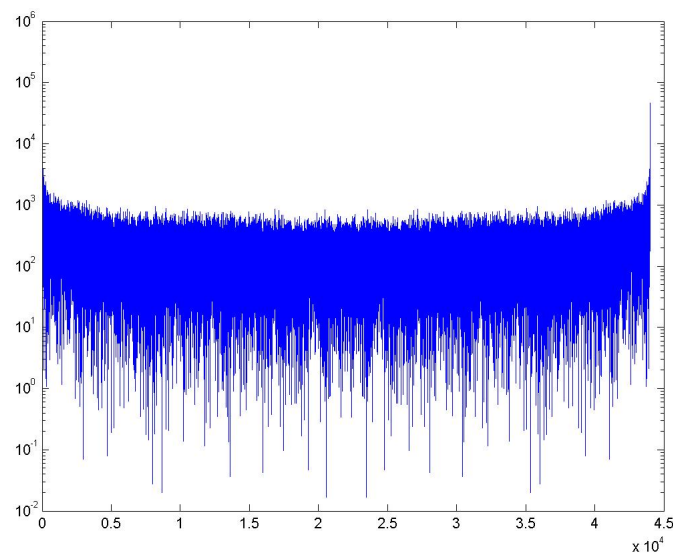


7.6 Analýza slova „lásky“

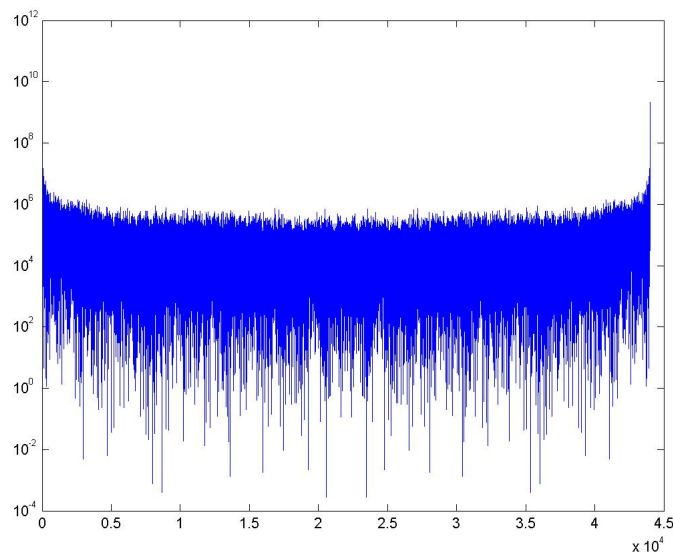
Obrázek 29: Signál zvukového záznamu $z = y(t)$ Obrázek 30: Fourierovo amplitudové spektrum $z = |FFT(y(t))|$ 

Obrázek 31: Power spektrum $z = |FFT(y(t))|^2$ Obrázek 32: Diskrétní spektrum $z = \log |FFT(y(t))|^2$ 

Obrázek 33: Amplitudové spektrum diskretního spektra $z = \left| FFT(\log |FFT(y(t))|^2) \right|$

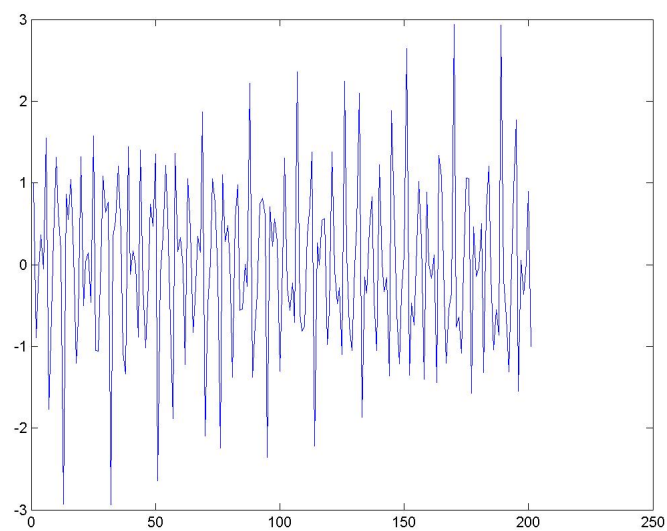


Obrázek 34: Power keprum $z = \left| FFT(\log |FFT(y(t))|^2) \right|^2$

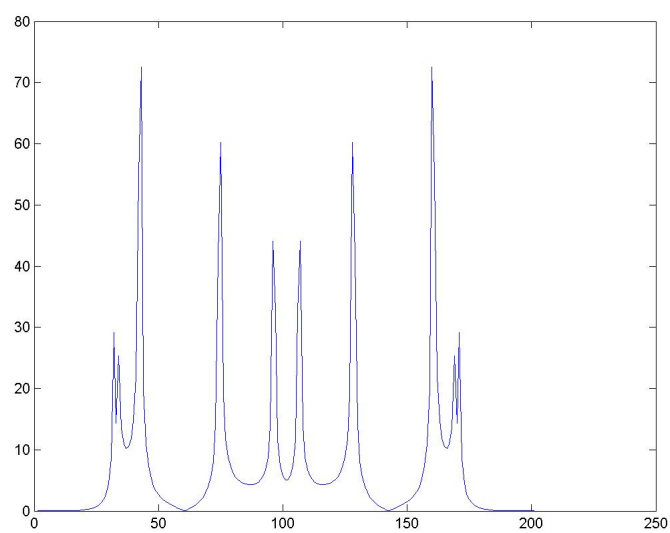


7.7 Aplikace Power kepra

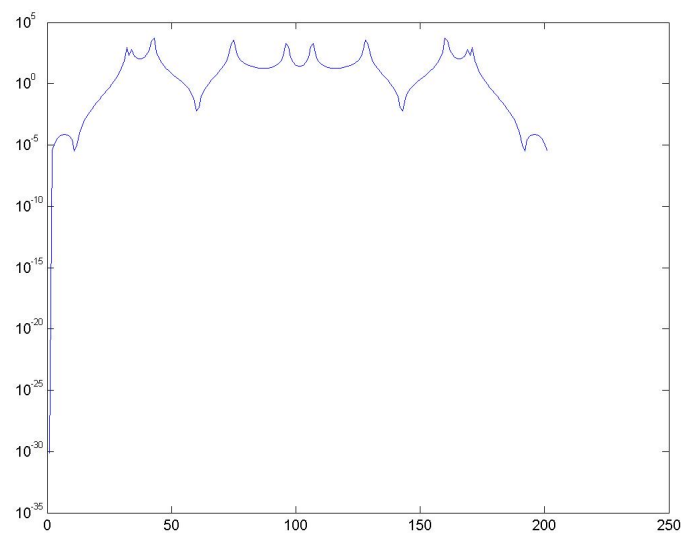
Obrázek 35: Signál $z = \sin(13t) + 0.8\sin(23t) + 0.6\sin(33t) + 0.4\sin(53t) + 0.2\sin(73t)$



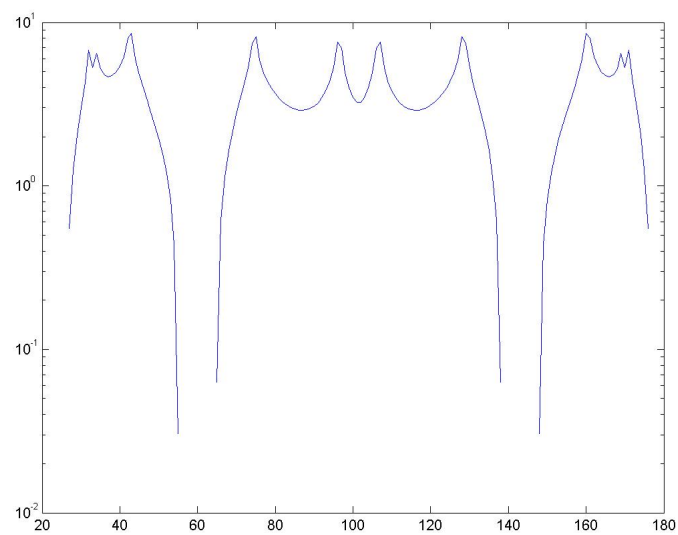
Obrázek 36: Fourierovo amplitudové spektrum $z = |FFT(y(t))|$



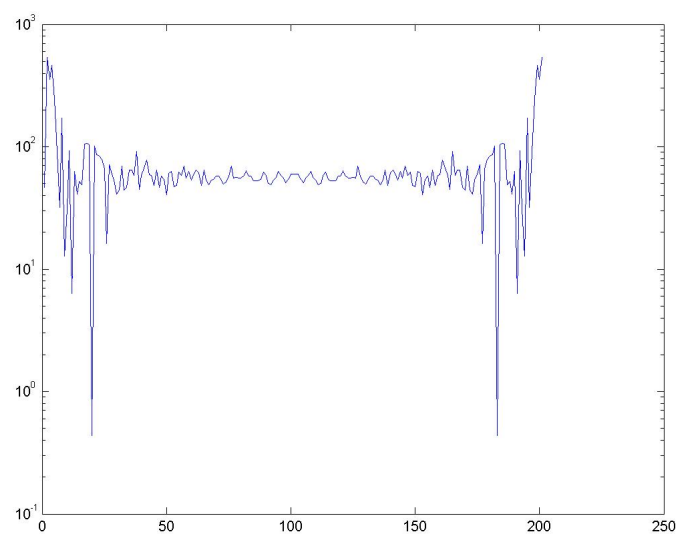
Obrázek 37: Power spektrum $z = |FFT(y(t))|^2$



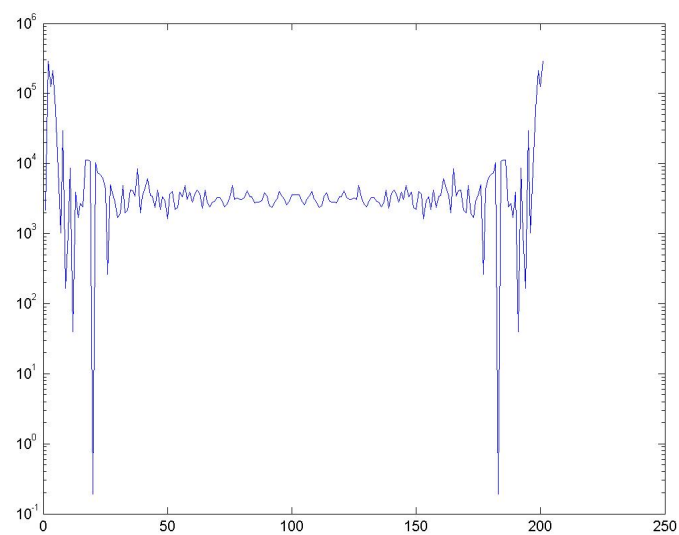
Obrázek 38: Diskrétní spektrum $z = \log |FFT(y(t))|^2$



Obrázek 39: Amplitudové spektrum diskrétního spektra $z = \left| FFT(\log |FFT(y(t))|^2) \right|$



Obrázek 40: Power keprum $z = \left| FFT(\log |FFT(y(t))|^2) \right|^2$



8 Závěr

Zjistili jsme, že technik pro rozpoznávání lidské řeči je hned několik. Porovnali jsme rychlosti různých implementací FT algoritmů. Pomocí kepstrální analýzy jsme zjistili, jak vypadají různá kepra různých souhlásek, samohlásek a slov. Technika rozpoznávání slov je však natolik složitá, že v budoucím vývoji bude třeba využít více sofistikovaných metod, které budou zaručovat vyšší pravděpodobnost úspěchu. Můžeme říci, že v této práci jsme detailněji rozebrali různé techniky rozpoznávání, různé algoritmy samotného výpočtu i liftrování. Tato problematika je velmi zajímavá a chtěli bychom se jí věnovat i v navazující diplomové práci.

9 Reference

- [1] Wikipedie: *Rychlá Fourierova transformace*
http://cs.wikipedia.org/wiki/Rychl%C3%A1_Fourierova_transformace
- [2] Wikipedie: *Fourierova transformace*
http://cs.wikipedia.org/wiki/Fourierova_transformace
- [3] tvorba webu.wz.cz: *FFT algoritmus*
<http://www.tvorba webu.wz.cz/faiirfe/help/fft.htm>
- [4] MUNI: *Paralelní FFT*
<http://www.fit.vutbr.cz/study/courses/PPP/public/prednasky/ppp-10.pdf>
- [5] RayMaps: *Daniel-Lanczosova metoda*
<http://www.raymaps.com/index.php/fast-fourier-transform-code/>
- [6] scribd.com *Short time kepsrum*
<http://www.scribd.com/doc/29323516/25/The-Short-Time-cepstrum>
- [7] MI21.VSB.CZ *Diskrétní transformace*
<http://mi21.vsb.cz/>
- [8] elektrorevue.cz *Kepstrální analýza*
<http://www.elektrorevue.cz/clanky/01048/index.html>
- [9] FFTW.org *Paralelní Fourierova transformace*
<http://www.fftw.org/parallel/parallel-fftw.html>
- [10] iitg.vlab.co.in *Cepstral analysis of speech*
<http://iitg.vlab.co.in/?sub=59&brch=4&sim=615&cnt=1>
- [11] indiana.edu *Danielson Lanczos algorithm*
<http://beige.ucs.indiana.edu/B673/node14.html>